(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 June 2002 (27.06.2002)

PCT

(10) International Publication Number
WO 02/50694 A1

(51) International Patent Classification[7]: G06F 15/173

(21) International Application Number: PCT/AU01/01629

(22) International Filing Date:
18 December 2001 (18.12.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PR 2154 18 December 2000 (18.12.2000) AU

(71) Applicant *(for all designated States except US)*: **TRAFFION TECHNOLOGIES PTY LTD** [AU/AU]; 278 City Road, South Melbourne, Victoria 3205 (AU).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CROSS, Paul, Gerard** [AU/AU]; 278 City Road, South Melbourne, Victoria 3205 (AU). **PRENDERGAST, Andrew** [AU/AU]; 278 City Road, South Melbourne, Victoria 3205 (AU).

(74) Agent: **FREEHILLS CARTER SMITH BEADLE**; 101 Collins Street, Melbourne, Victoria 3000 (AU).

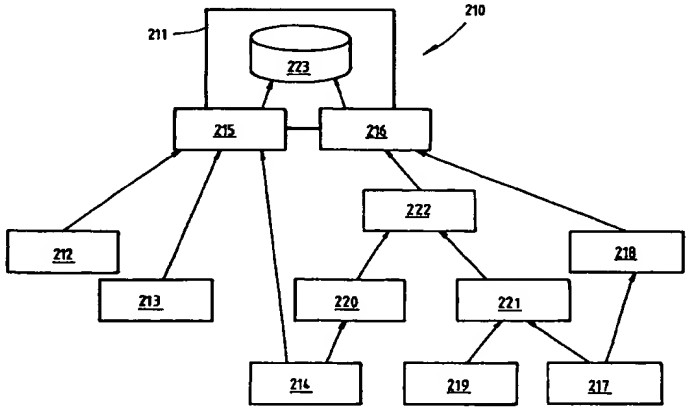
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD FOR TRACKING THE TRANSFER OF USER ATTENTION IN A COMPUTER NETWORK



(57) Abstract: A method of tracking browsing activity of a user in a computer network (210) including two or more independent computer network servers (220, 221, 2118) or server clusters, the user establishing a connection session with the computer network, the method including the steps of: receiving a user request for content located on a first of the network servers or server clusters; dynamically including a unique session identifier (session ID) in content served from that first network server or server cluster to the user during the connection session; transmitting the session identifier from the first (240) to a second (241) of the network servers or server clusters when a subsequent user request for content is made to the second network server or server cluster; transmitting an event message including at least that session identifier to a central session manager (211) upon the occurrence of a user browsing event during that connection session; and logging session information derived from the event messages in a session data structure (223).

WO 02/50694 A1

BEST AVAILABLE COPY

METHOD FOR TRACKING THE TRANSFER OF USER ATTENTION IN A COMPUTER NETWORK

The invention relates generally to a method for tracking the transfer of user attention in a computer network, such as the Internet.

Traditionally, an online marketer's ability to track the activity of a network user has been limited to the simple measurement of a user's access to and movement within an individual network site, without regard for the impact upstream activities have on downstream activities, as network users' within and across network sites controlled by the online marketer.

The business of buying and selling network user attention is a relationship between publisher and advertiser, typically undertaken according to one of the following a variety of business models.

Business relationships between publishers and advertisers are conducted according to an assortment of business terms tailored to each relationship, remuneration for which may or may not be financial. The advertiser or publisher may specify the technology platform used for each business relationship. Optionally, operators of network sites sit on the publisher side of some business relationships, while simultaneously conducting business as the advertiser in other relationships.

Consequently, publishers and advertisers are finding themselves using a variety of technology platforms based on different methodologies. The metrics generated by each of the methodologies may not be compatible; therefore comparing the performance of various relationships is difficult and often impossible.

Moreover, current methodologies do not facilitate the measuring of impact that upstream publishers have on downstream advertisers as network users pass through intermediary network sites.

Publishers and advertisers are also not able to conduct business according to multilevel payment models. These are models in which compensation is offered to any number of upstream publishers that are more than one network site detached from downstream advertisers.

It would therefore be desirable to provide a method of tracking the transfer of user attention in a computer network that alleviates or overcomes one or more problems existing methods.

It would also be desirable to provide a method of tracking the transfer of user attention in a computer network that is more accurate and reliable than existing methods across multiple network sites.

It would be desirable to provide a method of tracking the transfer of user attention in a computer network that was simple and convenient to implement and/or facilitated the analysis of captured network user data.

One aspect of the present invention provides a method of tracking browsing activity of a user in a computer network including two or more independent computer network servers or server clusters, the user establishing a connection session with the computer network, the method including the steps of:

receiving a user request for content located on a first of the network servers or server clusters;

dynamically including a unique session identifier in content served from that first network server or server cluster to the user during the connection session;

transmitting the session identifier from the first to a second of the network servers or server clusters when a subsequent user request for content is made to the second network server or server cluster;

transmitting an event message including at least that session identifier to a central session manager upon the occurrence of a user browsing event during that connection session; and

logging session information derived from the event messages in a

session data structure.

The event messages may be generated and delivered by the user. Alternatively, the event messages are generated and delivered by the network server providing content.

The event messages may be generated for session events, including any one or more of synchronisation event messages, impression events, result events, exposure events, click events and/or person events.

Another aspect of the present invention provides a method of tracking upstream and downstream traffic flow between multiple sites within a computer network; the method including the steps of:

at the central session manager, receiving site entrance and exit data collected upon the transferral of user attention between the multiple sites;

identifying network site relationships from correlations in the site entrance and exit data; and

establishing a sequence of cause-and-effect relationships between the multiple sites.

The site entrance and exit data may be captured by one or more tracking gates provided in one or more computer network servers hosting or forming part of a same network server cluster as the sites.

One or more of the tracking gates may be an exit tracking gate that dynamically includes a site identifier and/or session identifier in the link to the second site. The exit tracking gate may include a click sequence number or custom tracking data.

One or more of the tracking gates may be an entrance tracking gate that attempts to identify and record the site identifier and/or session identifier from data encoded in the user request. The data encoded in the user request may include any one or more of a URL, a cookie and/or a referrer.

One or more of the tracking gates may be a clickthrough tracking gate that records both first and second site identifier at the first site. The clickthrough gate may identify or create the session identifier, click sequence number and

custom tracking data from the first site.

The site entrance and exit data may be captured by session management based techniques. The session management based techniques may include analysing session data at the second site to identify entrance points from the first site.

The session management based techniques may include identifying external events directly resulting from an external hyperlink, and analysing external events to identify the first site. The external events may be analysed by extracting a session identifier, a click sequence number and/or custom tracking data from the first site. The external events may also be analysed by using any one or more of a flagged entrance algorithm, a sequencing based algorithm, or an internal identifier algorithm.

The session management based techniques may include applying a clicktrees tree index structure to identify session exit points.

The site entrance and exit data may be captured by session-less entrance detection means. The session-less entrance detection means may use an entrance recorded flag. The session-less entrance detection means may also use HTTP Host and Referrer Request Header Analysis.

The site entrance and exit data may be captured by generating transition identifiers.

The sequence of cause-and-effect relationships may be established by constructing a time-based clickstream tree index structure of use attention transfer.

A further aspect of the present invention provides a method of maintaining connection session state in a computer network including multiple sites, the user establishing a connection session with the computer network, information characterising the user being captured whenever the user interacts with each site to generate a user browsing signature; the method including the steps of:

at a central session manager, receiving the user browsing signature; and

deriving session information for the connection session from the transmitted browsing signatures.

The captured information includes information may identify a network presence of the user. Information identifying a network presence of the user may include the user's network address.

The computer network is preferably the Internet, and the user's network address may be an IP address.

The captured information may also include an data packet header. The data packet header may be an HTTP header. The data packet header may be a referrer request header, an accept language request header or a user agent request header.

The captured information may include user custom data. A source data string may be constructed from the captured information.

A hashing function may be applied to the source data string.

Yet another aspect of the present invention provides a method of tracking browsing activity of a user in a computer network including multiple sites, the user establishing a connection session with the computer network, impression events being generated when content is requested by the user from a network site, and result events being generated when the user completes a step towards a connection session result; the method including the step of:

at a central session manager, receiving messages representative of the impression and results events; and

interleaving impression events and result events to derive result progress path information for the user.

The results events may include at least result start, result report and result end events.

A still further aspect of the present invention provides a central session manager including a processing means and associated memory means, including code storage means for storing computer program code to cause the central sessions manager to carry out a method as previously described.

A further aspect of the present invention provides a computer program code for storage in memory means associated with processing means, the processing means and memory means forming part of a central session manager, the computer program code causing the central session manager to carry out a method as described previously.

The following description refers in more detail to the various steps and features of the present invention. To facilitate an understanding of the invention, reference is made in the description to the accompanying drawings where the method, system and system components of the invention are illustrated in preferred embodiments. It is to be appreciated that the present invention is not limited, however, to these exemplary steps and/or features.

In the drawings:

Figure 1 is schematic diagram illustrating a browser based user attention tracking system in accordance with the present invention;

Figure 2 is schematic diagram illustrating a server based user attention tracking system according to the present invention;

Figure 3 is a schematic diagram illustrating a combination of the systems illustrated in Figures 1 and 2;

Figure 4 is a more detailed schematic diagram of the central session manager forming part of the system illustrated in Figure 3;

Figure 5 is a schematic diagram illustrating exemplary ways in which network user attention is delivered from a first site to a second site in accordance with the invention;

Figures 6 and 7 are schematic diagrams illustrating the creation of connection sessions by network users on two network sites;

Figures 8 and 9 are schematic diagrams illustrating the creation of sessions by multiple network users across multiple network sites;

Figure 10 is a schematic flow chart illustrating the events occurring during a connection session resulting in a user joining a subscription based network site;

Figure 11 is a clicktrees tree index structure illustrating the interconnection between the events shown in Figure 10;

Figure 12 is a clicktrees tree index structure illustrating a variant of the structure shown in Figure 11;

Figure 13 is a schematic diagram illustrating the purchase of traffic from publisher sites by a broker site, and the on-selling of that traffic to advertiser sites;

Figure 14 is a schematic diagram illustrating the flow of traffic from a first group of network sites to a second group of network sites through a hub site;

Figure 15 is a detailed schematic diagram illustrating the flow of traffic through various hub sites to and from other network sites;

Figure 16 is a schematic diagram illustrating the flow of user attention between five different sites;

Figure 17 is a schematic diagram illustrating a clicktrees tree index structure representative of the flow of user attention illustrated in Figure 16;

Figure 18 is a variant of the clicktrees tree index structure shown in Figure 17;

Figure 19 is a schematic diagram illustrating the flow of information to and from various network sites through a central network hub;

Figures 20 and 21 are schematic diagrams illustrating the flow of user attention of two network users during connection session in which the network sites shown in Figure 19 are accessed;

Figure 22 is a schematic diagram illustrating a clicktrees tree index structure of the events occurring during the connection sessions shown in Figures 20 and 21;

Figure 23 is a schematic diagram illustrating various components of a central session manager and its interaction with various websites;

Figure 24 is a schematic diagram illustrating the data structure of a browser signature;

Figure 25 is a schematic diagram illustrating four impression events occurring during a connection session;

Figure 26 is a detailed schematic diagram showing the components and function of a central session manager forming part of the present invention;

Figure 27 is a general schematic diagram showing part of a network server and associated plugin operating in conjunction with the central session manager shown in Figure 26; and

Figure 28 is a detailed schematic diagram of the plugin of Figure 27.

There now follows a brief explanation of several terms used in the description:

(a) Browser Based Measurement

Upon receipt of content at a network user's client software ("browser"), the client software is typically expected to contact a measurement server and inform it of the content request. This client-initiated measurement technique is commonly referred to as Browser Based Measurement. Depending on the Browser Based Measurement implementation, a measurement directive may or may not be required to be included in the content served to the user to instruct the client to record a session event.

Passive Browser Based Measurement is essentially a tool that involves the placing of some particular piece of content, typically an image, onto a network page. Each time the network page is requested, the client also requests all of the files that are associated with that page including a measurement image that is used for measurement of viewing, or impressions, of that particular network page. To determine the results available from passive Browser Based Measurement, one would typically analyse a web server log file to determine the number of times that the measurement image has been called.

Active Browser Based Measurement is another sub-set of Browser Based Measurement. Unlike Passive Browser Based Measurement, before using Active Browser Based Measurement, network users are required to install an executable measurement add-on to their existing client software. This add-on is

responsible for capturing network user activity information and delivering it to a network server for processing.

(b) Server Based Measurement

When a client requests a document from a server, and the server subsequently delivers a document to the client, the server may optionally choose to record the request. The process of recording this request is known as Server Based Measurement. Typically, each time content is requested from the network server, a time stamped entry is typically recorded in a simple log file. Each entry included rudimentary information such as the document URL and the remote network address of the user.

An alternative to the basic log file is the storage of user activity information in a relational database management system, such as the widely distributed solution available with Microsoft's network servers such as Internet Information Server and Site Server.

(c) Log file Analysis

Log file analysis software is used to examine the log files and formulate a report. For example, log files might be analysed to determine the number of impressions and result events over a particular period.

(d) Session Management

Using a technique known as session management seemingly unrelated document requests are grouped together into a single session, thereby creating a list of related events. Without session management, it is difficult to determine whether a number of requests for various content over a network originate from a single user, or a number of users. For session management to function correctly, each user session is identified by way of a unique session identifier. Activity monitoring systems process and analyse the events for each network user's session with reference to the session identifier, thereby identifying relationships in a seemingly unrelated set of events.

Session Management can be applied to data collected via Browser Based or Server Based Measurement techniques, discussed above, provided that a technique has been applied at the data capture stage that results in a session identifier being included with the data.

(e) Session-less State Management

Session-less state management is a means of tracking relationships between seemingly unrelated network user events without the need for complex session identifier schemes used to identify individual user sessions.

Unlike session management, session-less state management involves grouping events associated with one or more user sessions under a single identifier. In reference to the publisher/advertiser model previously described, the identifier identifies a publisher's network site, and events occurring on an advertiser's network site are recorded with reference to a publisher's identifier.

(f) Ad-Serving

With traditional above the line media such as television, radio, print and outdoor, the advertising message remains static no matter how many people are exposed to it. Unlike traditional media, publishers can rotate advertising messages into and out of a single advertising space each time that it is viewed.

Ad-Serving technology facilitates this rotation. Every time a network user requests content containing advertising messages, an Ad-Server selects an appropriate advertisement and fills the space.

Ad-Servers are integrated into content one of three ways:

(i) Ad-Serving through Insertion

Integrated into the network server software, this form of Ad-Server modifies the content on the fly before delivering it to the network user.

(ii) Direct Ad-Serving

After the network user has received the content, special coding within the content instructs the network user's client to transfer part of the document request to an Ad-Server so the space may be filled.

(iii) Redirected Ad-Serving

It is possible to chain Ad-Servers so that the request transferred to the primary Ad-Server can be redirected to a second, third, fourth, and so on Ad-Server if so desired.

(g) Clickthrough Tracking

Clickthrough Tracking is used when the relationship between publisher/advertiser requires a measure of user attention caught, normally in terms of clicks sent into the advertiser's web site.

Publishers are expected to direct users (by way of a hyperlink) to the Clickthrough Tracking Gate. Encoded in the Clickthrough Tracking Gate's Universal Resource Locator (URL) is a publisher identifier and/or an advertiser identifier. When a network user clicks on the hyperlink, the request arrives at the Clickthrough Tracking Gate for processing. The server logs the request and then, according to the HTTP specification, sends a 3XX HTTP response code accompanied by a Location HTTP Response Header containing a copy of the advertiser's network address. On receipt of the information, the client automatically proceeds to the network address as specified in the Location Response Header without user intervention.

It is possible to chain Clickthrough Tracking Gates so that the request transferred to the primary Clickthrough Tracking Gate can be redirected to a second, third, fourth, and so on Clickthrough Tracking Gate if so desired.

(h) Referrer logging

Referrer logging is a technique applied when Clickthrough Tracking is not available but a measure of user attention caught by a publisher and directed into an advertiser's network site is still required. For example, a network site

maintainer generally has no control of search engine links and therefore is unable to direct network users through a Clickthrough Tracking Gate.

For referrer logging to function correctly, the application of some form of analysis is required to establish which page impression is associated with a network user entering a network site. Once established, the referrer field is extracted from the HTTP request data associated with the page impression previously identified.

The referrer field allows the client to specify, for the server's benefit, the address or URI of the resource from which a link was obtained. Given that the referrer field is an optional part of the HTTP specification, data loss should be expected.

In the Browser Based model shown in Figure 1, clients 200 to 202 deliver event data to an active sessions database with a session manager 203 after processing a content request served from a network server 204. Whether Active or Passive Browser Based Measurement is used, the network user's client software is expected to report to the Session Manager 203 separately from the document request.

In the Server Based model shown in Figure 2, a network server 205 delivers event data to an application running on the Session Manager 203. The Session Manager 203 can reside on the same machine as the Network Server 205, or a physically distinct machine.

In a preferred embodiment, the present invention is capable of accepting events from a network server via both a browser and server based interfaces. Regardless of the interface, a similar set of events are generated and uniformly applied to the same active sessions data structure. Whenever network users are encountered, whether via the Browser or Server Based model, their activity is translated by the relevant interface as a series of single events. The Session Manager maintains Session State as network users travel between various sites during connection sessions.

Network user activity is monitored according to an event model, which may include the exemplary events set out in Table 1.

Table 1

Example Event	Description
Session	This event is applied to the active sessions when a network user's session starts or ends.
Impression	This event is applied to the active sessions when a measure of user activity is required. Impression events document the path network users take through a network site from entrance to exit. Every time a document is requested from a network server, an impression event is triggered.
Result	Result events are applied to the active sessions when a network user achieves a quantifiable goal, financial or otherwise.
Exposure	Ad Exposure events are applied to the active sessions when an advertising message is delivered to a network user, in an attempt to capture network user attention.
Click	Click events are applied to the active session when network user attention has been successfully captured.
Person	Person events are generated to indicate, by way of persistent state, that a network user (or machine) has been identified by the system.

Session events describe an abstract concept in terms that can be easily dealt with by information systems. Session events can be applied to an active session record or a data structure. Session events include the events set out in Table 2.

Table 2

Example Event	Details

Session Start	On receipt of a session start event an entry under a specific session identifier is made in an active session structure under which further events can be recorded. When applying an event to the active session structure, under a specific session identifier, the system determines that that session identifier is not contained in the active session records, a new session is created records and a session start event is generated.
Session End	On receipt of a session end event, an entry in the active session structure identifiable by way of a session identifier is marked for removal. If, during the execution of a session end detection algorithm on the active session structure, it is determined that a session is no longer active then the session is marked for removal and a session end event is generated.
Session Heartbeat	The session heartbeat or synchronisation event allows an interconnected traffic manager to keep a real time list of all active sessions (and continue to pass the information along to other networked computers, as required) without interconnected traffic managers being confused of whether or not events have timed out and thereby preventing the tracking of sessions being prematurely terminated.
Session Map	On receipt of a session map event, one or more active sessions are combined so that all future references made under any of the identifiers refers to the same entry in the active sessions structure. Where it is determined that two or more session identifiers are being used to refer to a single network user's session, a session map event is generated indicating that an error correction adjustment should be applied.
Session Query	On receipt of a session query event, the entire contents of an active session are returned to the generator of the session query event.

Impression events are described differently between implementations of network site activity applications, for example site impressions, hits, and page requests. Impression events are different to exposure events. Exposure events are also known as banner impressions or simply impressions.

Impression event data may be used to record the start of a session and record how a network user arrived at a network site. That data is also used to calculate traffic indicators, including how many network users have traversed a network site; how often users return, how long users spend on the site, and how many pages they reviewed at the site.

Result events can be categorised as events occurring when the user achieves a measurable goal, financial or otherwise. Broadly speaking, result events generally fall into the categories set out in Table 3.

Table 3

Result Type	Example
Lead	A user joins a mailing list or creates an account on your site
Subscription	A user subscribes to a pay-for-access site that is billed monthly
Softgood	A user purchases digital goods that are delivered via the net
Hardgood	A user purchases hard goods that are delivered offline

In each of these situations, there might be a confirmation page at the end of the process that informs the user that they have joined the mailing list, that their digital goods are available online at a specific URL; that their goods have been shipped, etc.

Ad exposure events represent an attempt to capture network user attention, while click events represent the capture of network user attention. It is common to see a combination of both exposure and click events within a single session.

A person event is generated when a network user is encountered. Two types of person event exist – new person and existing person. Person events can be implemented by way of identifying a network user's machine through the use of persistent cookies, and storing habit information, which facilitates the generation of existing person and new person events at appropriate intervals based on past contact with the network user.

A single document or other content may be delivered to a network user by a network server, and may contain one or more advertising messages, trigger one or more result events; and/or trigger one or more person events. Therefore, if delivery of that content generates an impression event, the impression event can have a relationship with one or more secondary exposure, result, or person events.

Figure 3 shows generally a system 210 for tracking the transfer of user attention within a computer network in accordance with the present invention. In the system 210, a central session manager 211 accepts event messages directly from network users 212, 213 and 214 via a browser interface 215. The central session manager 211 is also capable of accepting event messages via a server based interface 216. On the serving of content to a network user 217, a network server 218 may provide an event message to the server interface 216. Alternatively, when content is served from network servers 220 and 221 to network users 214 or 217 or to a network user 219 an event message, or other information representative of an event, may be forwarded to a network session manager 222, for forwarding to the server interface 216 of the central session manager 211.

Regardless of the interface used or the means by which the event message or other information corresponding to an event is generated and transported to the central session manager 211, a similar set of events are generated and uniformly applied to the same active sessions data structure. Regardless of the interfaces available, network user activity is translated into events, which are then applied to an active sessions data structure. Browser

interfaces manager this translation on the receiver request from a network user, while server interfaces process events generated externally of the central session manager 211. As soon as Figure 4, browser interfaces may include an ad-server 224, Clickthrough gate 225 and Measurement gate 226. Server interfaces may include an events gate 227 and structures gate 228.

The central session manager 211 maintains a single active session data structure across one or more browser or server based interfaces. The central session manager 211 records one or more events with regards to a single session identifier, a process known as maintaining state. Several forms of state may be maintained, as set out in Table 4 below.

Table 4

Type of State	Description
Site State	State within a network site is required to group events generated by a network user while visiting a network site.
Cross Site State	Cross-site state involves linking one or more groups of events arranged through the maintenance of site state. In this way, when cross-site state is maintained one can track the movement of a specific user as they move between network sites.
Persistent State	Persistent state refers to the maintenance of state between sessions. Persistent state is normally dealt with in terms of a single network user accessing a single network site. Example applications of persistent state are Person Events and Custom Tracking Data (see above).

A wide variety of situations exist whereby a network user is directed to a second site with or without the aid of a Clickthrough Tracking Gate and/or an Ad-Server. Figure 5 illustrates four exemplary ways in which network user attention is delivered from a first site to a second site, based on the optional presence of the Ad-Server 224 or Clickthrough Tracking Gate 225 shown in Figure 4, in circumstances where a known connection session already exists on the first site.

These cases are respectively referenced 242 to 245. In all four cases, a session identifier is passed from a first network site 240 to a second network site 241 by way of URL token, referrer token or cookie. However, in situations where none of these options exists, digital foot printing may be used to match two seemingly unrelated sessions.

When a session is created on the first site, the first site is preferably identifiable by way of a site ID, which is then passed to the second site by way, for example, of URL token or referrer token.

In cases where a Clickthrough Tracking Gate does not exist, then a clear identification - in terms of a click event - that network user attention has been delivered to a network site is not generated. As an alternative, various algorithms can be applied at the second network site to identify, the first network site (if any).

Each of these algorithms assume that one or more events exist within a session at the second network site 241 and that the algorithm is applied to the set of events at the second network site 241 for the purpose of identifying those events which contain information relating to the first site 240. The event is then analysed and the first site 241 identified, either in terms of a session identifier relevant to the first site or in terms of a site identifier that identifies the first site.

Figure 6 illustrates an arrangement where five network users, respectively referenced 270 to 274, access two network sites 275 and 276. Each of the network users 270 to 274 establishes a separate connection session, respectively referenced 277 to 282. The network sites 275 and 276 form part of a computer network and may be maintained on independent computer network servers or server clusters. When each network user 270 to 274 accesses one or other of the sites, a separate connection session is established. Figure 7 illustrates the manner in which events generated by each network user 270 to 274 are represented in terms of an active session data structure maintained in the active sessions database 223 of the central session manager 211. In this figure,

impression events are represented by "Imp", whilst result events are represented by "Res". The events occurring during each session are represented in a logical clicktree tree index structure representative of the browsing activity of each user within each connection session.

Whilst Figures 6 and 7 describe an arrangement in which each session state is maintained within a single site, Figures 8 and 9 represent the manner in which the present invention is able to maintain multiple cross site relationships within a single active sessions data structure. In this example, three network users, respectively referenced 300, 301 and 302, simultaneously access three network sites, respectively referenced 303, 304 and 305. In this scenario, the second network user 301 has accessed all three network sites 303 to 305 during a single connection session 306. The network users 300 and 302 respectively establish connection sessions 307 and 308. In Figure 9 "Exp" represents an exposure event, while "click" represents a click event.

The exemplary data structure shows that during connection session 306, the network user 301 responded to an advertising message appearing on site A, briefly visited site B then proceeded to a clearly definable result event at site C. Clearly, the result at site C was a direct result of an advertising exposure appearing on site A.

Much of the data maintained by active sessions data base 223 is in terms of publisher/advertiser relationships, whereby, for any given publisher/advertiser relationship, a detailed session structure describing interaction within and between sites is available. A further dimension to the data is possible in terms of custom tracking data. For each session, a persistent set of custom tracking data can be maintained for each network user visiting each network site. If a new publisher/advertiser relationship delivers the same network user to the same network site later, the existing persistent custom tracking data can be overwritten and replaced with new custom tracking data.

Custom Tracking Data may be defined by publisher and/or advertiser and normally consists of one or more identifiers that describe the relationship in further detail. For example, a placement identifier can be incorporated into the custom tracking data to describe where on a page an advertising message has appeared; a further placement identifier may be provided to identify where in a network site an advertising message appeared; and an advertisement identifier might be used to identify the exact advertising message presented to a network user.

As has been described previously, results events are generated when a network user achieves a quantifiable goal, financial or otherwise. A confirmation page is normally presented to the user after the result has been achieved and, at the same time, a result event is generated.

For example, a non-financial result event occurs when a user joins a mailing list or creates an account on a network site. In these situations, a confirmation page at the end of the process typically informs the user that they have joined the mailing list or supplies their username and password respectively. After the result event is generated, a sales representative might contact the network user later and attempt to sell them one or more products.

Another example is when a user subscribes to a pay-for-access site that is billed monthly. In this case, a confirmation page at the end of the process informs the user of their password and might detail how to cancel the subscription in the future. Several months and billing cycles after the result is generated, the network user might cancel their subscription.

When a user purchases hard goods that are delivered offline, a confirmation page is presented with a receipt and an estimated delivery time. After the event has been generated, a fulfilment house may determine that part or all of the order is out of stock and issue a refund accordingly.

In all of these examples, actions may occur well after the initial session has finished that, while a side effect of the session, are not directly related to the original session.

Moreover, the result event mentioned does not provide insight into any multistage registration or purchase processes that have led to the generation of a result. Considering that many result confirmation pages involve a multistage linear process leading up to the confirmation page (and result event), a need exists for measurement of dropout rates when the linear process commences but is not completed. A fine-grained measure of when a result process has begun, how far a user has been through the process, and whether the process was completed would be desirable. In addition, a means of detecting dropouts would be beneficial.

In one or more embodiments, the present invention clearly identifies when a result begins, when the user has moved closer to or further away from completing the result, and when a result ultimately ends. Exemplary result events are shown in Table 5.

Table 5

Event Name	Details
Result Start Event	The Result Start event informs the central session manager that a network user has begun a potentially result-generating process.
Result Report Event	As the user moves through a result generating process, result report events are optionally generated at relevant stages informing the central session manager that a progress meter should be updated.
Result End Event	The Result End event is comparable to other result measurement systems. When used without Result Start and Result Stop events, the outcome is identical to that of other result measurement systems. In regards to the improved system, a Result End event informs the central session manager that a result (that was previously started) has now been completed.

Result start events represent a network user commencing a multistage result generation process. Parameters to a result start event may include a Result ID, Result Name, and Maximum Progress Limit, as explained below:

(a) Result ID

When a network site expects network users to complete one or more result generating processes of the same type during the same session, the network site operator provides a result ID for each result generating process. The result ID must not be repeated in the same session. For example, an online trading web site expects users of network sites to place multiple bids in a session, sometimes simultaneously. Whenever the process of placing a bid begins, a Result Start event is generated containing a bid number. Subsequent Result Report and Result End events must also include the same result ID.

(b) Result Name

In situations where a network site offers multiple types of results, the start event for each type of result is given a mutually exclusive name, which is also to be used in subsequent result report and stop events. For example, a magazine publisher's network site might offer its publications electronically after a user completes a result generating process to create a username/password for accessing the publications. A separate result generating process is offered for users requiring delivery of printed via postal mail. The events associated with these results might be named "online subscription" and "offline subscription" respectively.

(c) Maximum Progress Limit

It has been determined to be useful analyse where in the process a user has dropped out. If this is to be analysed in terms of what percentage of the result process was successfully completed, the maximum size of the result process can be provided as part of the Result Start event. When specifying a Maximum Progress Limit, the maximum number of linear steps required to

reach a Result End event not including the initial Result Start event, can be defined.

Result Report events are used to describe at which stage in a result generating process the network user is positioned. The Result ID and Result Name are only included in a Result Report event if the accompanying Result Start event also included a Result ID and/or Result Name. In addition to the Result ID and Result Name parameters, Result Report events also include Progress Interval and Interval Description parameters, as explained below:

(a) Progress Interval

The numeric progress interval describes which stage the network user is currently at in relation to the Progress Limit specified during a Result Start event.

(b) Interval Description

The interval description is a human readable description of the current stage. In situations where the progress interval is not sufficient for identifying the network user's current stage within the process, An interval description can be used to describe the stage in question.

A Result Stop event marks the successful completion of a result generating process. A result event can be used by itself, or in conjunction with Result Start and Result Report events.

No event is provided for informing the central session manager that a user has dropped out of a result generating process – this is intentional. If a session end occurs and the session contains unfinished result generating processes (in other words, one or more Result Start events without an associated Result End event), they can be recorded as a result that has dropped out.

Figure 10 illustrates a practical example of the foregoing. Consider the example of a user joining a subscription based network site, where access is

provided to network users at a monthly rate of $29.95. Figure 10 illustrates the information architecture of the process required to subscribe.

In this process, two payment options are provided at step 350 – the user has a choice of entering a credit card number at step 351 or alternatively calling a 1-900 number at step 352. The credit card option is further complicated, in that if the credit card is rejected the first time at step 353, a second attempt is provided at step 354. The transaction is either subsequently denied at step 355, or approved at step 356. In the 1-900 option, entry of a code is required at step 357. Unsuccessful entry of the code results in rejection at step 358, or else approval of the transaction at step 356. Following approval, a confirmation page is served at step 359.

The following sessions are provided in the context of a scenario in which three network users attempted to subscribe using the above subscription process, of which only one succeeds. A list of events for each session is provided, along with a Clicktree tree index data structure for each session describing how each event occurring during that session is related.

From the example shown in Table 6 and illustrated in Figure 11, a linear process can be seen to have been followed by the user, eventually resulting in a Result End event. In this situation, the user did not drop out of the Result process and therefore very little extra information has been achieved by including the additional result events.

Table 6

	Event	Details
1	Impression	Choose Payment Method
	Result Start	No Result ID; No Name; Progress Limit = 5
2	Impression	Call 1-900 for code

	Result Report	Progress Interval = 1
3	Impression	Enter Code
	Result Report	Progress Interval = 2
4	Impression	Transaction Approved
	Result Report	Progress Interval = 3; Description = Approved
5	Impression	Confirmation Page
	Result End	

In the example shown in Table 7 and Figure 12, no result end occurred, however result dropout information is still available. In addition, two attempts were made by the user to subscribe – first with a credit card, and the second time via the 1-900 facility. Note that the progress interval counter was reset to one during the transition from the fifth to the sixth impression – in this example, both the credit card and 1-900 payment options are considered part of the same process.

Table 7

	Event	Details
1	Impression	Choose Payment Method
	Result Start	No Result ID; No Name; Progress Limit = 5
2	Impression	Enter Credit Card Info
	Result Report	Progress Interval = 1
3	Impression	Transaction Denied
	Result Report	Progress Interval = 2; Description = Credit Card Rejected

4'	Impression	ReEnter Credit Card
	Result Report	Progress Interval = 3
5	Impression	Transaction Denied
	Result Report	Progress Interval = 4 Description = Credit Card Rejected
6	Impression	Call 1-900 for code
	Result Report	Progress Interval = 1
7	Impression	Enter Code
	Result Report	Progress Interval = 2
8	Impression	Invalid Code
	Result Report	Progress Interval = 3 Description = 900 Code Rejected

In existing Advertiser Centric systems, publishers send traffic into an advertiser and reports on publisher performance are made available from the advertiser's perspective. The inverse applies to existing Publisher Centric systems, where a publisher sends traffic into multiple advertisers and reports on advertiser performance are made available from the publisher's perspective.

Considering that most publishers deal with a number of advertisers and vice versa, and each publisher may be using a publisher and/or advertiser centric system to manage the flow of traffic, publishers and advertisers often find themselves using a variety of publisher centric or advertiser centric systems to essentially analyse the traffic flowing into or out of their network site.

Consider the example shown in Figure 13 whereby a broker site 400 purchases traffic from publisher sites 401 to 403 then re-sells the traffic to advertisers 404 to 406.

Traffic flowing from any of the publishers can be expected to arrive at any of the advertisers. From the broker's perspective, it would be most useful if data was available to this effect, describing the impact traffic flowing from upstream sites has on the traffic flowing into downstream sites. This data could then be used in future purchasing decisions or the broker may choose to structure remuneration of upstream publishers according to the traffic and results generated at downstream advertiser sites.

In the above described example, one could say that the broker site is simultaneously playing the role of publisher and advertiser.

In one all network sites can play the role of publisher and advertiser. In a traffic-trading environment, the aim is to acquire traffic from other sites, convert it into results, and then re-sell the traffic to other network sites.

Applying the traffic-trading model to the broker example, the result is now as shown in Figure 14. There are no longer any publishers or advertisers – just network sites. What was the broker site is now a hub site 420, in that any traffic that flows from sites 421 to 427 must pass through hub site 470.

The example shown in Figure 15 illustrates a typical situation where a group of loosely connected hub sites 430 to 433 collect traffic from a wide variety of network sites 434 to 448, then re-distribute the traffic amongst themselves in an attempt to extract the highest amount of results from the traffic.

In the context of the present invention, traffic is traded in units of network user attention. Traffic flow exists when a network user's attention is transferred from one network site to another. When network user's attention flows from one network site to another, it may be measured according to a variety of metrics, the most common metric used is a click. Optionally, advertisers, in consideration of traffic flowing from the publisher to the advertiser, remunerate publishers according to the traffic flow.

Tracking upstream/downstream traffic flows can involve identifying the transfer of network user's attention between sites as a sequence of cause and

effect relationships. In some situations, the flow of network user attention from a first to second to third network site is clearly definable, for example, when click events are generated connecting the first network site to the second, and the second to the third. In other situations, the relationship is not as clear, for example, when a first network site mentions that a network user should visit a second network site, and as a result, a network user suddenly appears at a second network site.

A number of observations can be made which facilitate the upstream/downstream tracking of network user attention. Firstly, if a network user can be independently identified at the first and second network sites, then the transferring of, the network user's attention from the first network site to the second network site is a single operation identifiable by the first or second network site without the need for measurement data from the second or first network site respectively. In other words, a network user exiting from one network site can be later matched against an entrance at another network site. If the network user cannot be independently identified at the first and second network site, then the connection between activity at the first and second network site may be identifiable using the data recorded during the transition from first to second site. In other words, the data recorded during the transition must include details of the first and second network site, including session identifiers at the first and second network site.

Secondly, when a network user exits a first network site, a second network site is entered. A network user can exit a first network site multiple times; only a single second network site is entered each time. A network user can enter a second network site multiple times; only a single (one) first network site is exited each time. An exception to this is that a network user may exit a network site without entering another network site. In this case, the network user's attention has reached its destination.

Thirdly, if the transfer of a network users attention from a first to a second network site can be identified; and the transfer of a network users

attention from a second to a third network site is may be identified; then the first network site affected the network user attention delivered to the third network site by way of the second network site.

These three observations can then be used to analyse a series of exits and entrances across a group of network sites to establish, for each network user, the sequence of cause-and-effect relationships between each network site.

In order to track upstream/downstream traffic flow it is desirable to identify where and when a network user has entered and exited a network site. Depending on the technique used, either an entrance into a network site is identified; an exit from a network site is identified; or both an exit from a first network site with a corresponding entrance into a second network site is identified. In addition, a variety of additional data relating to the first and/or second network site can be collected, including a site identifier, a session identifier, a click sequence number and custom tracking data.

The techniques used to track user attention flow include gate based techniques, including the use of exit gates. An exit gate is used to identify a network user leaving a network site. When a user chooses to exit, they are first directed to an exit gate, which in turn records the network user leaving a first network site. The exit gate then directs the user to a second network site. The first network site may be identifiable in the information recorded by the exit gate. Optionally, when directing the network user to the second network site, the exit gate may choose to incorporate a site identifier and/or session identifier from the first network, in the link to the second network site. A click sequence number or custom tracking data may also be included.

Entrance tracking gates may also be used. An entrance gate is the inverse of an exit gate – it is used to identify a network user entering a network site. Users enter via an entrance gate, which in turn records the network user entering the second network site. The entrance gate then directs the user into a specific section of the second network site. The second network site may be

identifiable in the information recorded by the entrance gate. Optionally, the entrance gate can attempt to identify and record the site and/or session identifier from the first network site by way of data encoded in the URL, referrer or cookies. A click sequence number and custom tracking data may also be identified.

Another gate that may be used in a Clickthrough Tracking Gate. Clickthrough gates combine the functionality of exit and entrance gates. When leaving the first network site, the network user is directed to a Clickthrough Tracking Gate. The Clickthrough Tracking Gate identifies and records both the first and second network site, and optionally identifies a session identifier, click sequence number and custom tracking data from the first network site. In session management based environments, the Clickthrough Tracking Gate may also create a session identifier for use by the second network site.

The Clickthrough Tracking Gate then directs the network user to the second network site, optionally incorporating a site identifier from the first network or a session identifier in the link to the second network site. A click sequence number or custom tracking data may also be included.

The techniques used to track user attention may also include session management based techniques. For example, session data may be analysed to identify entrance points. When an Entrance or Clickthrough Tracking Gate does not exist, then a clear identification that network user attention has been delivered to a second network site is not available. An alternative is to analyse detailed session data for the second network site to identify which parts of the detailed session data includes entrance data. The algorithm used to conduct the analysis may assume that one or more events exist within a session at a second network site and that the algorithm is applied to the set of events at the second network site for the purpose of identifying those events that contain information relating to the first site. The following algorithms work on a common principal that there should be a differentiation between events that are a direct result of a

hyperlink external to the network site (in other words, from a first network site) as compared with events that are a result of a hyperlink within the network site. After identifying one or more external events, they are further analysed and the first site further identified. Optionally, a session identifier, click sequence number, and user definable tracking data from the first site may be extracted from the event data as part of the further analysis.

The first algorithm is a flagged entrance algorithm. When network users are directed to a second network site, some form of flag is supplied to the second network site to highlight events. Session data is then searched for events containing the flag. Flagged events are then analysed to identify the first network site, (if any). For example, a hyperlink presented on the first network site contains a flag indicating to the second network site that a session identifier from a first network site is being provided.

Another algorithm is a sequencing based algorithm. This algorithm involves application of a sequencing technique to session data. After sequencing, event data is arranged in such a way that the first event can be identified. One means of sequencing a session is time based, where events are sorted based on when the occurred. An alternate arrangement includes using Clicktree based sequencing, where events are arranged according to matching previous and current click sequence numbers. After sequencing is applied to the events from a second network site's session, the first event is assumed to contain information relating to the first network site, if any. In the case of applying Clicktree based sequencing, multiple first events may be identified and analysed.

A third algorithm is an internal identifier algorithm. According to the common principal described above, there needs to be a differentiation between events that are a direct result of a hyperlink external to the network site as against events that are a result of a hyperlink within the network site. In line with this principal, session data is analysed to determine those events that are a

direct result of hyperlink within the network site. After internal events have been identified, the remaining events are assumed to be the direct result of an external hyperlink, and as such are analysed to identify the first network site, if any. For example, a site maintainer may choose to add an "internal" flag to all hyperlinks within the site that announces that the next event is the result of an internal hyperlink.

Another example does not require site-specific modifications of each link, but rather is based on analysis of the referrer and host HTTP request headers. Given that the host header identifies the hostname of the current network site, if the hostname from the host header matches the hostname from the referrer header, the assumption can be made that the request is the result of an internal hyperlink. In the case of a blank HTTP request header, the assumption is made that the event is the result of an external hyperlink.

A further session management based technique used to track user attention includes applying a Clicktrees tree index structure to identify session exit points. Given the nature of hypermedia systems, it is difficult to identify when a network user has left a network site without the use of an Exit or Clickthrough Tracking Gate. However, when click sequence numbers are included with session data, and a click sequence number from a first network site is transferred to a second network site, an clear link between the exit of a first network site to the entrance at a second network site can be established.

The techniques used to track user attention may also include session-less entrance detection techniques. The above-described techniques have all required Session Management or Gates to function. Various options exist, however, which do not require a gate or session management to function. Even in situations where Gates and/or Session Management exist, the following options may be used due to site specific limitations on passing of site identifiers, session identifiers, click sequence numbers and custom tracking data between

first and second network sites. These options include the use of an Entrance Recorded Flag.

Whenever a request to a network site occurs, the request may be checked to see if a flag representing the network user entering the network site has already been recorded. If the flag does not exist, the current event is analysed for identifying a first network site that may have potentially sent the network user into the second (current) network site. After analysis, the flag can then be attached to the network user. This can be in the form of a cookie attached to the network user or, alternatively, hyperlinks that reference the current network site might include the flag.

Another option includes HTTP Host & Referrer Request Header Analysis. Given that the HTTP referrer request header identifies the network address of the previous resource that referenced the current resource, in situations where a first network site has directed a network user into a second network site, the HTTP referrer header contains the full network address of the first network site. Given that the HTTP host header identifies the hostname of the current network site, if the hostname from the host header does not match the hostname from the referrer header, it can be assumed that the current request is the result of a first network site directing a network user into a second (current) network site. In this case, the current request at the second network site can be is analysed so as to identify the first network site.

The techniques used to track user attention may also include generating transition identifiers. Each time a network user's attention is transferred from a first network site to a second network site, a unique transition identifier may be generated. The first and second network site and any gates in between can then track exit and entrance events with reference to the transition identifier. In this way, detailed data relating to both the exit and entrance pages is made available, even in situations where a single network user has entered or exited a network site multiple times. While by itself not a means of detecting entrance, this

technique can be used to apply session management entrance (and exit) detection techniques to session-less environments.

As previously described, when network user attention is transferred from a first network site to a second network, it can be recorded with reference to the first and/or second network site. The second part of tracking upstream/downstream traffic flow may be to analyse these records to identify correlation between records documenting when network user attention left a first network site, and records that document network user attention delivered to a second network site.

Depending on the means by which the transfer of attention is recorded, establishing these 1:1 relationships between first and second network sites can occur at different times. The earlier the relationship is established, the more robust the data. Accordingly, a clear relationship between a network user exiting one network site and the same network user entering another network site can be defined. Strong relationships are identified at the time network user attention was transferred. For example, in the case of Clickthrough Tracking Gates, a single piece of data can be generated at the time of transferring network user attention that identifies both the first and second network sites.

In the case of using Transition Identifiers, again the identifier can be generated at the time of transferring network user attention and as such and , the relationship between the first and second network site is very strong.

In the case of transferring a site identifier, session identifier, and possibly a click sequence number from a first network site to a second network site, again the relationship between the first and second network site is very strong.

However, a weak relationship will be defined when two separate pieces of data representing an entrance and an exit are joined together for the purpose of defining a network user exiting one network site and the same network user entering another network site. Weak relationships are identified once the two separate entrance and exit pieces of data are joined together. For example, when user attention is transferred from a first network site directly to a second

network site, and Browser Based Measurement at the second network site identifies the same user entering the second network site from a first network site, a weak relationship between the first and second network site is established.

When neither the exit or entrance can be clearly defined no clear relationship will be established. However, an assumption can be made that a network user appearing on a second network site was the result of that network user accessing a first network site. This type of relationship can be determined, for example, after no more data is available relating to the network user's current activity and a sequencing algorithm is applied to establish relationships between network sites. Consider the example of a network user appearing at a first network site, then a few minutes later appears at a second network site. Although no information exists that clearly identifies the relationship between the first and second site, an assumption can be made, based on sequencing events according to the time at which they occurred, that the first network site has some relationship with a network user delivered to a second network site.

From the perspective of a single user, after both strong and weak relationships have been identified, a sequence of cause-and-effect relationships can be established.

In the following example illustrated in Figure 16, a network user travels between five network sites 500 and 504, two of which (referenced 500 and 503) do not have any measurement software installed. A list of network site relationships can be created as follows:

Site 500 to 501: A clearly definable relationship between the first and second site exists due to the use of a Clickthrough Tracking Gate.

Site 501 to 502: A weak connection between the second and third site has been established. Browser Based Measurement at the third network site detected that the network user arrived by way of a session at the second network site.

Site 502 to 503: No knowledge of the transition from the third to fourth site is available.

Only measurement data is collected for the activity at network site 504, and no relationship established. This measurement data is however collected with reference to the same user that accessed sites 500 through 502.

After establishing links between sites, part of the story is now complete. Arranging the data according to the active sessions tree index data structure shown in Figure 17, we see the relationships between sites 500 to 502 clearly identified. Events are arranged according to time-based clickstream sequencing so the earliest events are at the top and the latest events at the bottom. Site 500 does not have a session, however it is still included according to the data supplied via the Clickthrough Tracking Gate. Site 503 is not included as no data was collected, thereby creating a gap.

Once all the data has been collected, the clickstream sequencing can be analysed to discover that the first event at site 504 occurred shortly after the last event at site 502. An unclear relationship between sites 502 and 504 is filled in. The upstream/downstream relationship between sites is now site 500 to site 501 to site 502 to site 504.

In general, the upstream/downstream flow of network user attention can be defined according to the first entrance point for each session. However, the active sessions data structure maintained by the central session manager 211 serves as a useful way to programmatically deal with the problem of multiple entrance and exit points. The example scenario shown in Figure 18 involves a single network user accessing four network sites 502 to 523. Represented in this figure is that site 521 delivered the network user to both sites 522 and 523. The network user also returned to site 521 after responding to an advertisement at site 522, and subsequently generated a result.

If applying the general rule of thumb that the upstream/downstream flow of network user attention should be defined according to the first entrance point for each session, one would determine that the result generated at site 521 was a result of the network user being delivered from site 520.

However, detailed analysis of session activity can be applied for the purpose of determining which network sites where most responsible for driving results. Consider the result generated at site 521 in the above example. If only the first entrance to site 521 was analysed, it is determined that site 520 is responsible for the result at site 521. No allowance has been made for the fact that the network user responded to an advertisement at site 522 before generating the result at site 520.

Analysis of the detailed interaction between sites 521 and 522 though, enables one to determine that the network user attention flowed sequentially through sites 500, 501, 502 and 501 before achieving the results. After detailed analysis, it becomes clear that site 500 played only a minor role in driving the result and that site 522 was the initial source of the result.

A more detailed scenario will now be considered with reference to Figure 19. In the following scenario, network site 540 acts as a hub and directs traffic from the sites 546 - 548 into sites 542 - 544. In addition, traffic flows freely between sites 541 and 545 through site 540. Figures 20 and 21 illustrate examples for two network users traversing the network of network sites shown in Figure 19 – Users X and Y.

In Figure 20, User X begins at network site 548, then at step 600 travels to network site 540 via a Clickthrough Tracking Gate. Once at network site 540, the User X travels directly to network site 541 at step 601, then at step 602 returns to network site 540 via another Clickthrough Tracking Gate. After returning to network site 540, the network user X finally travels back to network site 542 at step 603.

In Figure 21, user Y begins at network site 547, then at step 610 travels to network site 540 via a Clickthrough Tracking Gate. Once at network site 540, the User Y travels directly to network site 543 at step 611.

Figure 22 provides a more detailed view of the transfer of user attention of User X. User X begins at the first network site 548. Given the lack of measurement at the first network site; only exposure events are recorded. After the Browser Based Ad Server Interface delivers four advertising messages, the network user responds and clicks on an advertising message. The network user's attention is then transferred from the first network site to the second network site 540.

The Browser Based Ad Server Interface, by way of session cookies, can maintain site state while at the first network site 540. Alternatively, a Browser Signature for User X can be generated and checked against a list of active Browser Signatures for the first network site 540. To ensure that browser-based session cookies are not re-used mistakenly across network sites, each request to the Browser Based Ad Server Interface may include the site identifier, for example:

http://www.optitrack.com/adserver?site=9

Given the presence of the Browser Based Clickthrough Tracking Gate, cross-site session state between the first site 548 and second site 540 involves two stages. The first stage involves maintaining cross-site state between the first network site 548 and the Browser Base Clickthrough Tracking Gate by way of session cookies or a browser signature. Again, the request to the Browser Based Clickthrough Tracking Gate may include the site identifier, for example:

http://www.optitrack.com/clickthrough?fromsite=9&tosite=1

The second stage involves maintaining cross-site state between the Browser Based Clickthrough Tracking Gate and the second network site 540. Prior to directing the user to the second network site 540, the Clickthrough Tracking Gate creates a new session entry in the Active Sessions tree index structure maintained in the central session manager 211. For the second

network site 540, in this case Session B, and a click event recorded with reference to the new session. The new session identifier is then encoded into a URL in a form recognisable to the second network site 540, and the network user directed to that URL as follows:

http://www.site1.com/index.cfm?session=B

After network user attention is transferred from the first network site 548 to the second network site 540, Server Based Measurement at the second network site 540 detects the supplied session identifier, and tracks impression events with reference to the supplied session identifier (as opposed to generating a new session identifier). These impression events are delivered to the central session manager 211 by way of the Server Based Event Interface 216, and include information that identifies the second network site 540 and the session identifier at the second network site 540. Using this information, the central session manager 211 can easily determine where in the active session structure to insert the generated events.

From a page received from the second network site 540, the network user clicks on the following link:

http://www.site2.com/index.html?site=1

After clicking on the link, the network user is transferred from the second network site 540 to the third network site 541. On arrival at the second network site 540, Server Based Measurement carried out by the central session manager 211 creates a new session identifier and supplies events to the Server Based Event Interface 216, again containing identifiers for the third network site and session. In addition to the site and session identifiers for the third network site 541, included in the impression is the second network site's identifier retrieved from the URL site 548 above, along with a session identifier for the second network site 540, which was extracted from the referrer HTTP header. The central session manager 211 then uses the previously described Flagged Event, First Event, or External Event algorithms to determine which impression should

contain the second network site's session and site identifiers and use this data to maintain cross-site state.

While at the third network site 541, an advertising message is delivered to the network user from the Browser Based Interface 215. Again, site state must be maintained, but this time its cross-interface – the session and site identifier used at the third network site 541 must be supplied to the Browser Based Interface 215. One option is to encode the data in the request to the Browser Based Ad Server, as follows:

http://www.optitrack.com/adserver?site=2&session=C

When the request arrives at the Browser Based Interface 215, an exposure event is generated and stored with reference to the session at the third network site 541. The Browser Based Interface 215 also creates a session cookie for later reference.

After receiving the advertising message, the network User X responds once again by clicking on the link. The two-stage process required to deliver cross-site state via a Browser Based Clickthrough Gate is once again required. The first stage involves maintaining cross-site state between the third network site 541 and the Browser Base Clickthrough Tracking Gate by way of session cookies or a browser signature. Again, the request to the Browser Based Clickthrough Tracking Gate may include the site identifier, for example:

http://www.optitrack.com/clickthrough?fromsite=2

The second stage involves maintaining cross-site state after the network User X leaves the Browser Based Clickthrough Tracking Gate. Prior to directing the user to the second network site 541, the Clickthrough Tracking Gate locates an existing session entry in the Active Sessions tree index structure 223 for the destination network site 540, in this case Session B and a click event recorded with reference to the new session. The new session identifier is then encoded into a URL in a form recognisable to the second network site 540, and the network user directed to that URL, as follows:

http://www.site1.com/index.cfm?session=B

After network user attention is transferred from the third network site 541 back to the second network site 540, Server Based Measurement at the second network site 540 either detects the supplied session identifier, and tracks impression events with reference to the supplied session identifier (as opposed to generating a new session identifier) or detects an existing cookie. Impression events are delivered to the central session manager 211 by way of a Server Based Event Interface 216, and include information that identifies the second network site 540 and the session identifier at the second network site 540. Using this information, the central session manager 211 can easily determine where in the active session structure to insert the generated events.

From a page received from the second network site 540, the network user once again clicks on a link, as follows:

http://www.site3.com/index.html?site=1

After clicking on the link, the network User X is transferred from the second network site 540 to the fourth network site 542. On arrival at the fourth network site 542, Server Based Measurement creates a new session identifier (session D) and supplies events to the central session manager 211 via the Server Based Event Interface 216, again containing identifiers for the fourth network site 542 and session. In addition to the site and session identifiers for the fourth network site 542, included in the impression is the second network site's identifier retrieved from the URL (site 548 above), along with a session identifier for the second network site 540, which was extracted from the referrer HTTP header. The central session manager 211 then uses the Flagged Event, First Event, or External Event algorithms to determine which impression should contain the second network site's session and site identifiers and use this data to maintain cross-site state.

Figure 23 illustrates the integration of the session manager 211 and the cross-linking sessions as a user travels from site to site to thereby enable, upstream and downstream traffic analysis to become possible. As the page for Web Site 650 is served to the user, a Content Server module 651 selects an

appropriate Web Site 652 creative from the Web Site 650/Web Site 652 relationship to serve from a database 653 at step 654, logs an impression event at step 655, which is then placed into the session list in database 223. This logging records the display of Web Site 652's creative to the user. The creative is then served at step 655 to the user for display on Web Site 650's web page, and the Browser Based Measurement module 671 detects an impression at step 657 and so logs an impression event at step 658 in the session list at step 659.

If the user clicks on Web Site 652's creative, which is contained on Web Site 650, then an HTTP request is sent to a Redirect sub-module 670 at step 672, which is logged in the session manager 211 at step 673 and recorded in the session list at step 659. The link to the next page, here Web Site 652, is retrieved at step 674 from the database and the user is automatically redirected at step 675 to page on Web Site 652.

As with Web Site 650, the Content Server module 651 determines the appropriate Web Site 676 creative from the Web Site 652/Web Site 676 relationship that is stored in the database 653, and which is to be served as part of Web Site 652 at step 677. As Web Site 652's page is displayed, an impression event for that page is detected at step 678 and logged at step 658 and stored 659 in the session list.

If the user clicks on Web Site 676's creative, which is contained on Web Site 652, then an HTTP request is sent to the Redirect sub-module at step 679, which is logged in the session manager at step 673 and recorded the session list at step 659. The link to the next page, now Web Site 676, is retrieved at step 674 from the database and the user is automatically redirected at step 680 to page on Web Site 676.

As with Web Sites 650 and 652, the Content Server module 651 may determine that there is an appropriate creative from some relationship where Web Site 676 is an affiliate, and which is stored in the database 653 that can be served as part of Web Site 676 at step 681. As Web Site 676's page is

displayed, an impression event for that page is detected at step 682 and logged at step 658 and stored at step 659 in the session list.

From this example, it can be seen that the concentration of event recording provided by the above-described integration leads to a new ability to be able to attribute the arrival of traffic at Web Site 676 to not only Web Site 652, but also Web Site 650. The present invention allows the tracking of a user across multiple web sites and not just the movement from one site to another. Following the process shown above, this integration allows tracking of a user or session to occur across an infinite number of sites, so long as each site is connected to a system according to the present invention.

Various additional means of maintaining cross-site state may also be used by the present invention. These include the use of browser signatures. A browser signature is a short-term network user identifier that functions via the HTTP protocol without the need for cookies. A browser signature can be used most commonly as a short-term replacement for session and/or user ID's, where cookies or other means of maintaining Session State does not exist.

Consider for a moment that a single network user's access of one or more network sites generally tends to include the same or similar IP addresses for the duration of the session. Consider also that given the wide variety of network client software, the slight variations in the software between each version, and the different way in which each browser is configured. A browser signature involves gathering up all this information and generating a short-term user ID from the data. To generate a browser signature, a string containing a variety of information pertinent to the network user is constructed. A hashing function is then applied to the string. The result is a numeric identifier by which the network user can be identified.

Construction of the source string is an important part of generating a Browser Signature. There can be no limit to the size of the source string – it can be as long as is required to fit the source data. As shown in Figure 24, the first 24-bits or three bytes may contain the first three bytes of the network user's IP

address 700. After the IP address, a selection of HTTP header 701 from the network user's client software may be appended to the source string. After the selection of HTTP headers 702, optional custom data can be appended. For example, a network site ID could be incorporated if you required different Browser Signatures between network sites.

The selection of HTTP headers chosen to incorporate into the source string is important to the application in question. Effectively a browser signature identifies a network user using their specific IP address and HTTP headers. The fewer HTTP headers that are incorporated into the source string, the higher the likelihood that two network users will simultaneously be allocated the same browser signature. The more HTTP headers incorporated into the source string, the higher the likelihood that one network user will simultaneously have multiple browser signatures.

The minimum headers include are the contents of the HTTP referrer request header, the HTTP accept language request header, and the HTTP user agent request header. Further request headers can be chosen on the basis that they would not change during a session, thereby generating a new Browser Signature.

The selection of custom data is based on the application in question. In situations where network user activity is being tracked for a number of network sites simultaneously, adding the network site identifier to the source string ensures that browser signatures remain unique for each network site.

After constructing a source string, a hashing function is applied to the source string, resulting in a numeric browser signature. The choice of hashing function is based on the requirement that whenever the exact same source data is passed to the hashing function, the same browser signature is returned. In addition, if the source string changes only slightly, a different hash is generated. Finally, the numeric value returned by the hashing function should be at least a

32-bit number. An ideal example hashing function is applying a CRC-32 to the source string.

To maintain Session State using browser signatures, every time the network user is encountered, a browser signature is regenerated. The browser signature is then used in place of a session identifier. To maintain cross-site Session State, when a user is first encountered on a network site, a browser signature is generated. If the user travels to a second network site and a browser signature at the second network site matches a browser signature from a first network site, a relationship between the network user's session at the first network site and the network user's session at the second network site is created.

The Referrer request-header field in HTTP requests allows the client to specify, for the server's benefit, the URI of the resource from which the Request-URI was obtained. Figure 25 illustrates four impression events 720 to 723 occurring as user attention is transferred between two sites 724 and 725. For each impression event, both the request URI and the HTTP referrer are shown. At impression 722, the transfer from the first site 724 to the second network site 725 includes the first network site's site identifier in the URL (site=9). In addition, the referrer contains the session identifier of the first site 724. The second site 725 is capable of determining the session and site identifiers from the first network site by analysing both the requested URL and the HTTP request header. Therefore, the referrer header can be used as an aid to maintaining cross-site state.

A detailed embodiment of the invention will now be described with reference to Figures 26 to 28. The central session manager and network server described hereafter form part of the systems 210 of Figure 3, and implement the above-described methods and techniques. The central session manager 800 shown in Figure 28 includes a Server Measurement Module 801, Browser-Based Measurement module 802; and Content Serving module, which includes the Redirect sub-module 804.

The Server Measurement Module 801 is responsible for receiving and processing various event messages that relate to a user's navigational behaviour across a network 3. The event messages may originate from one or more information servers that are connected to the network 3, and which run a plugin illustrated in Figures 27 and 28 as part of the information server configuration. The Server Measurement Module 801, may exist independently of the other modules that make up the system illustrated in Figure 26.

The Server Measurement Module 801 operates by listening for incoming event messages 1 that are received from across the network 3. When a message arrives at the Server Measurement Module 801 at step 4 then the module accesses stored data in a database 5 to retrieve information that relates to the merchant that is specified in the incoming event message at step 6.

Once the module has retrieved the relevant merchant's information from the system data store 5, then the module looks at the incoming message at step 7 to determine if this is a time synchronisation event message. Time synchronisation messages are used to synchronise the system with the merchant's information system, which may be physically located in a different time zone. The synchronisation event helps to keep both the merchant's online system synchronised with the traffic measurement system. If this is a synchronisation event, and the module identifies a difference between the time specified in the synchronisation message and the system time, then the system adjusts the amount of delay between the time of each event received from the merchant and the system clock at step 8. If this is a time synchronisation message, which has been received by the module then no further processing of that incoming message will occur at step 9.

However, if the module determines at step 7 that this is not a synchronisation event message then the module begins to record information that is contained in the incoming event message, and which is forwarded from the relevant plugin located on the remote web server on the network 3.

Each event message that arrives at the module 801 in may, have multiple different session IDs for each event. The Server Measurement Module 801 processes each different session ID in turn at steps 10 and 15.

The module begins by examining the first session ID that is contained in the incoming event message that was received at step 10. The central session manager 800 then examines the list of sessions that are currently being tracked to determine if that first session ID already has an entry in the session list, signifying that this session is currently being tracked at step 11 . This involves a query of the list of sessions that are currently being tracked at step 12.

If this first session ID is contained within the session list, then the Server Measurement Module 801 will store the event information at step 13 into a list of events 14 that relate to this particular session.

Once the Server Measurement Module 801 has processed the first session ID contained in the incoming event message, the module then examines the incoming message to determine if the message contains any other session IDs to which this particular event will also apply at step 15.

If there are further session IDs to be processed, then the next session ID is examined at step 10, otherwise there is no further processing of this event message at step 16. This next session ID is then also checked at step 11 against the session list 12 to determine if that session ID is contained within the session list that is currently being maintained.

If the Server Measurement Module 801 determines at step 11 the list of this session IDs 12 does not contain the session ID that is currently being considered, then the Server Measurement Module 801 creates a new session entry in the session list 12 and then performs a session mapping. Session mapping occurs when one incoming event messages contains two or more different valid session IDs. It is clear in such an instance that the same user has, for various reasons relating to the technical aspects of some networks, been allocated multiple session IDs. The logical step is then to map the events that occur in each of the different session IDs as being events that relate to a single

session 17. This may also be accomplished by mapping one session ID that has been allocated to the tracked user to refer also to the different, but mapped, session IDs.

As with the previous session ID, if this session ID exists at step 11 in the current list of sessions 12, or if it did not exist at step 11 but has now been created at step 17 and mapped, then the module 801 will store at step 13 the event for this particular session into list of events 14.

This process will continue while there are still session IDs in the incoming message that have not yet been handled. When there are no more session IDs in the incoming event message to be processed at step 15 then further processing of this incoming event message terminates at step 16.

Browser Based Measurement involves the tracking of a user's movement across a network 3, by examining the requests made of the central session manager 800 server to serve a one-by-one pixel GIF, those requests having originated from a particular user that is connected to the network 3. HTML code, which is included in the network pages that are served to the user, directs the user's browsing software or client, to request the serving of a particular file from the server.

A web server at the site 2 that is connected to the network 3 initially receives the incoming request for the file, and that web server then communicates that request to the Browser Based Measurement module 802 at step 18 via a Fast CGI interface. Typically, the file requested is a one-by-one pixel GIF and the incoming request will be an HTTP request.

When a request for the one-by-one pixel GIF is received at the Browser Based Measurement module 802 at step 18, then the module extracts data required for further processing from that request at step 19.

One field that is extracted from the incoming HTTP request is the merchant for whom this one-by-one pixel GIF has been requested to allow tracking. Information relating to this particular merchant is then retrieved at step 20 from the data store 5.

The Browser Based module 802 then considers at step 21 whether the merchant specified in the incoming request is contained in data store 5. If the system is currently handling the merchant specified in the incoming request then that merchant will be stored in the data store 5. If the Browser Based module 802 detects at step 21 that this merchant is not currently contained in the data store 5, then the server will serve an error GIF at step 22 back to the requesting user's browser software and all further processing is terminated at step 9.

If the Browser Based module 802 detects at step 21 that this merchant is currently in the data store, then the module examines at step 23 whether this incoming request is part of an event that signifies that an impression or a transaction has taken place. An impression event is simply that a particular piece of content, such as an image, has been sent to, and therefore probably viewed by, the user. A transaction can be a financial or non-financial transaction and includes the purchase of an item, or the subscription to a mailing catalogue.

If the Browser Based module 802 detects at step 23 that this request is not part an impression or a result event, then the server will serve an error GIF at step 22 back to the requesting user's browser software and all further processing is terminated at step 9.

If the Browser Based module 802 detects at step 23 that this request is part of either an impression or a transaction event, then the module calculates a digital footprint for the requesting user at step 24.

The Browser Based module 802 then considers at step 25 whether there is a session ID that is contained in the incoming request, which is possibly stored in a cookie that exists on the user's computer, or client.

If the Browser Based module 802 detects at step 25 that this request contains a Session ID that corresponds to a session that is already stored in the session list 12, then the server will serve that same session ID back to the user at step 27 in a cookie.

If the Browser Based module 802 detects at step 25 that this request does not contain a session ID, or contains a session ID that does not correspond to a session that is already stored in the session list 12, then the server will create a new session ID for this user at step 26, insert that session into the session list 12, and then serve that new session ID back to the user 27, usually in a cookie. The Browser Based module then refers to the session list 12 to determine if this particular merchant that is referenced by this incoming request has a session ID in the list of sessions that is maintained at step 28. If this particular merchant does not have an entry in the session list for the current session ID, then a new session is created for the merchant and stored in the list at step 29. If there is already a session list entry for this merchant at step 28, or a new session has been created in the list at step 29, then the Browser Based module 802 determines whether this incoming request is part of an impression event at step 30. If this is an impression event then the module logs at step 31 this impression event in the list of events 14. If this is not an impression event at step 30, or it is an impression event and the event has now been logged into the event list at step 31, then the Browser Based module considers this incoming HTTP request to determine if this event is part of a result at step 32.

Where the incoming request is part of a result event that has been initiated by the network user, then the Browser Based module 802 checks at step 33 whether there is a cookie that is temporarily stored on the user's computer that contains a digital footprint. A digital footprint is a value that represents, to the finest level of accuracy possible, this particular network page being viewed by this particular user.

If this is not a result event at step 32 or if this is a result event, but there is a digital footprint stored on the user's computer that matches the digital footprint of the current result event at step 33, then the Browser Based Measurement module 802 serves the result one-by-one pixel GIF at step 34 and then terminates further processing at step 35.

If the digital footprint matches a digital footprint stored on the user's computer then it is apparent that this user has just previously visited this result page and another result should not be recorded. This storage of the digital footprint is used to ensure that if the user, for example, re-navigated back through the current page then the system will not count that navigation as another result. If the user did re-navigate to this current page as part of a result then the system changes the digital footprint so that the new result will be recorded as a different result to the original transaction.

If this is a result event but the digital footprint of this current page does not match a digital footprint that is currently stored on the user's computer, then the Browser Based module 802 logs at step 36 the result event into the list of events 14 and then serves the digital footprint of this current page at step 37 to the user in the form of a cookie at step 34.

Once the result has been logged at step 36 and a cookie containing the current digital footprint at step 37 has been served back to the user, then the Browser Based module 802 serves the result one-by-one pixel GIF at step 34 to the user and terminates further processing at step 35.

The third module in the central session manager 800 is the Content Server module 803, which is responsible for providing and tracking specific content to users located on the network 3.

As with the Browser Based Measurement module 802, Optitrack's Content Server module 803 receives messages from a co-located web server 2 via a Fast CGI interface that operates between the module and the web server at step 38.

The first step taken by the Content Server module 803 is to consider at step 39 whether this incoming request is a request from a network user's computer that requests some particular content to be served to that user's computer for display to the user. Typically, the user's computer makes such requests because of code that is placed into the HTML embedded in affiliates network pages.

If this is a request from a user connected to the network 3 for content to be served to the user, then the next consideration made by the module 803 is whether this is a request to serve an advertisement, or creative, of some type to the network user at step 40.

If this is a request to serve some particular creative to a network user then the Content Server module considers for which merchant this creative has been requested. Following that, the module retrieves that particular merchant's information at step 41 from the data store 5 that is maintained by the server.

Next, the Content Server module 803 examines the incoming request to determine two additional values used to appropriately match the incoming request to an appropriate creative from the group of available creatives. Those two additional values are the affiliate whose page contains the HTML code requesting the creative, and the size of the creative that is required to be displayed on the affiliates network page at step.

The Content Server module 803 then considers the relationship between affiliates and the range of creatives that are available to that affiliate at step 43. If this incoming request for a particular creative is not a valid request, because of an invalid creative size or invalid affiliate, then the Content Server Module 803 serves an error GIF at step to be displayed on the affiliates page, and further processing is terminated at step 45.

If, however, this is a valid request for a particular creative at step 43, then the Content Server module 803 goes on to consider all of the relationships between the merchant and the affiliate. In other words, the module then examines each of the merchant's advertising campaigns that involve this affiliate at step 46. This process essentially involves of examination of each of the merchant's campaigns that are underway where this particular affiliate is participating at step 46, and then examining each creative in each such campaign at step 47 to determine if any of those creatives match the incoming request at step 48. Practically, this may be accomplished by looping through each of the merchant's campaigns where the affiliate is participating at step 46

and 50, and then also looping through each creative set to be used in each of those campaigns at steps 47 and 49, until a suitable creative is found at step 48.

Logically, the module 803 examines the first of the merchant's campaigns where this affiliate is participating at step 46. For that first campaign, the module 803 checks each creative that is set to be used in that campaign at step 47 to determine if any creative matches the incoming request at step 48. If, after checking each creative set to be used in this first campaign, the module 803 finds that no creative match the affiliate's incoming request then the module checks another of the merchant's campaigns where this affiliate is participating. If after examining all of the creatives at step 49 in all of the merchant's campaigns at step 50 where this affiliate is set to participate the module does not find a match at step 48 for the incoming request, then an error GIF is served at step 44 to be displayed as part of the affiliates network page, and further processing is terminated step 45.

If the Content Server module 803 detects a creative that exists in a campaign that is a part of the merchant/affiliate relationship at step 48 then the Content Server module 803 crosschecks the matched creative at step 51 to determine whether this is, in fact the correct creative to serve for display as part of the affiliate's network page. This is achieved by crossmatching the creative/campaign relationship and the relevant campaign against the merchant/affiliate relationship. Each should be a member of the corresponding relationship.

If this cross matching detects that the selected creative should not be served as part of the affiliates network page at step 51, then the Content Server module serves an error GIF at step 44 and further processing terminates at step 45.

If this is a valid creative at step 51 that has been selected to be served to the network user then the Content Server module 803 creates a digital footprint at step 52 that is a unique identifier that represents this particular network user having viewed this particular creative to be served as part of this particular

affiliate page. This digital footprint may be effectively used as an impression event for the selected creative. That digital footprint is then added at step 53 to the event list 14. Once the footprint, or impression event, has been added to the event list, then the creative is served to the network user at step 54 as part of the affiliates network page and further processing is terminated at step 45.

The Redirect sub-module 804 of the Content Server module 803 is typically used to detect instances where a user has clicked on a particular piece of content on an affiliates network page, perhaps a creative served by the Content Server module 803 as outlined above, and has moved to another page on the network. An example of such an event is that a network user clicks on a merchant's advertising banner that is displayed on an affiliates network page, whereupon the affiliate would wish to know that a user has navigated from the affiliates page to the merchant's network site.

Initially, if the Content Server module 803 determines that an incoming request is not a request to serve content, but is actually a request to redirect the network user to another location at step 39, then the Content Server module 803 passes the incoming request to the Redirection sub-module 804 so that it may record the event and then redirect the user to the specified redirect location at step 55.

The next step that the sub-module 804 undertakes when redirecting a user to another location is to retrieve at step 56 the details of the affiliate, who is specified in the incoming request from the user. It is the affiliate's network page that contained the redirect link and the sub-module will be able to determine the affiliate based on details that are embedded in the redirect link.

Once the relevant affiliate has been determined , the affiliate's details are then retrieved at step 56 from data store 5.

The sub-module 804 then considers whether the affiliate specified in the incoming request is a valid affiliate at step 57, who is known to the server and stored in the data store 5.

If the sub-module 804 is unable to detect that the affiliate specified in the incoming request is a affiliate known at step 57, then the network user is redirected to an error page at step 58 and further processing terminates at step 59.

If the sub-module 804 detects that the affiliate specified is known, then the sub-module 804 determines at step 60 whether the incoming request also contains a creative identifier (an Ad ID), which specifies the particular creative that has been clicked, the campaign to which that creative belongs, and the merchant who is running that particular campaign.

If the incoming request does not contain an Ad ID then the sub-module generates a digital footprint at step 61, which uniquely identifies this particular creative, the campaign to which the creative belongs, and the merchant who is running the campaign. In essence, this digital footprint is instead of an Ad ID.

Once the digital footprint has been generated at 61, then the sub-module 804 performs a look-up at step 62 of the digital footprint list 34 which matches the digital footprint value to a creative identifier, a campaign identifier and a merchant identifier. The sub-module 804 then performs a crosscheck to verify the creative, campaign and merchant identifiers are valid at step 64.

If the incoming request for a redirect contains an Ad ID at step 60, the sub-module 804 extracts at step 65, from that Ad ID, identifiers for the creative that was clicked, the campaign to which that creative belongs, and the merchant who is running that campaign. The sub-module 804 then performs a crosscheck to verify the creative, campaign and merchant identifiers are valid at step 64 and match. In other words, the crosscheck at step 64 ensures that the creative does belong to the specified campaign, and that the merchant is actually the merchant who is running the specified campaign.

The sub-module 804 then generates a digital footprint for those incoming requests that contain an Ad ID, and uses the digital footprint already generated for those that do not, to create an event that corresponds to a click of the creative on the affiliate's network page, or click event at step 66.

The sub-module 804 then looks up the list of sessions 12 to determine if the session ID contained in the incoming redirect request is contained within the list at step 67.

If the session ID contained in the incoming redirect request does not exist in the session list, then the sub-module 804 creates a new session ID at step 68 and inserts that session into list of sessions 12. If the sub-module 804 finds that the session ID exists at step 67 in the list of sessions 12, or the session ID has just been created at step 68 in the session list 12, the session ID is served back to the network user, in the form of a cookie, for future reference at step 69.

Once the session ID has been served back to the network user, the click event is then inserted at step 70 into the list of events 14.

The content that is to be served to the network user is then parsed to insert tokens into that content at step 71. The redirect page is then served to the network user at step 72 and further processing is terminated at step 73.

Figure 27 shows generally a network user or client 101 connected to a network server 151 in this example by the Internet 102. The network server 151 is representative of the network servers 220, 221 and 218 shown in Figure 3, and acts to serve content stored in a storage media 116 to the client 101. The network server 151 includes network server software 157, the functionality of which is determined by a network server plugin 158 acting in conjunction with an information server engine 154 to serve content stored on the storage media 116 to the client 101. The network server plugin 158 includes a session ID detection module 155, a synchronisation module 153 and a filter module 152.

The operation of the network server 151 will now be briefly described with relation to Figure 28. Initially, the client 101 makes a request to receive content located on the network server 151. This request is intercepted by the session ID module 155 and analysed at step 103. At step 104, the session ID module 155 determines whether a token has been encoded within the HTTP request sent by the client 101. Next, the session ID module 155 determines at step 105 whether a cookie stored on the client 101 has set a cookie value, acting

as a session identifier, within the HTTP request. In the next step 106, the session ID module 155 determines whether a session identifier can be derived from the referrer field in the HTTP request header.

If the session ID module 155 determines that a session identifier can be determined at any one or more of steps 104 to 106, the session identifier will be temporarily stored in a memory storage device 107.

If the module 155 is unable to positive identify the user as a user whose browsing activity is currently being tracked by matching the detected session identifier with previously generated and stored session identifiers, the session ID module 55 will trigger a flagged impression event at step 109.

Alternatively, if the session ID module 155 is able to identify the user as one that is currently being tracked at step 108, a non-flagged impression event will be triggered at step 110. At step 111, flagged and non-flagged impression events are placed into a queue in a memory device 112. Subsequent event messages are added to the memory device 112, and the contents of that memory device may be subsequently retrieved by the traffic manager module 156 to link session events within a same connection session.

Once the session ID module 155 has stored an event message in the memory device 112, the content request is passed at step 113 to the information server engine 154. Once the server request is received by the information server engine 154 at step 114, the information server engine 154 retrieves the requested content stored in the storage media 116 at step 115, and streams or otherwise transmits this retrieved content at step 117 to the filter module 152.

At step 118, the content forwarded from the information server engine 154 is received at step 118, and subsequently examined at step 119 to determine if the content is part of a sale, subscription, query or other result event. If the content does contain something to signify that the content is part a result event at step 120, the filter module 152 raises a result event message at step 121, and transmits this event message for storage in a queue maintained in the memory device 112.

The filter module 152 then dynamically alters the content at step 122 to be served to the user, in order that the browsing activity of the user may be tracked during the current connection session. Information that is dynamically included into the content at step 122 is obtained either from counter that exist in a memory device 123, or from configuration files 124, which may typically be maintained locally on the network server 151. Once the content has been dynamically altered at step 122, the content is transmitted at step 125 across the Internet 102 to the client 101 for viewing by the user.

The synchronisation module 153 generates a set of synchronisation event messages to allow for temporal stamping of information flowing from the plugin 158. Synchronisation event messages are initiated at step 126 by the generation at regular intervals of a sync signal, which are sent to the memory device 112 at step 127 either as a timesync event message 128 or a server ping message 129. The purpose of the timesync event message 128 is to mark the place in the queue maintained in the memory device 112 when a particular time occurred. For example, if the queue maintained in the memory device 112 was backlogged by 30 minutes worth of processing, then the timesync signal will still allow interfacing software to record the viewing transactions with some form of time date or other temporal data. At predetermined intervals the server ping messages 129 are may be placed in the queue maintained in the memory device 112 for each website or source of content active on the webserver 157. The purpose of the server ping message 129 is to signal to interfacing software that the network server is still active.

A central session manager 160 interfaces with the network server plugin 58, and is responsible for processing the event messages generated by an event message transmission module 161 forming part of the network server plugin 58. The module 161 acts at step 162 to transmit event messages stored in the data structure 12 to the central session manager 160 to enable tracking of user attention at a central location.

The network session manager 222 shown in Figure 3 may include the same modules as that shown in Figure 28, with the addition of a traffic manager module to enable the centralised tracking of user activity in sites maintained on servers (such as those referenced 220 and 221) in communication with the network session manager 222. Having developed a session list of events occurring at those sites, the network session manager 222 then transmits the event messages in the session list to the central session manager 211.

From the foregoing numerous features and benefits become apparent. As the above-described system consolidates the three once disparate competencies of ad-serving, measurement and tracking modules, the appropriate payments to be made to an affiliate based on a number of known predefined events are able to be determined.

A centralised control console for a number of publishers that provides the mechanism for any publisher to buy or sell, or buy and sell user attention is also provided.

Numerous merchant banners can be rotated through the Ad-Server and provide accurate accounting that allows affiliates to receive one payment at the end of the month. A centralised control console also allows publishers to choose to download and serve banners if they wish, and also allows publishers that are acting as merchants and/or affiliates to form user attention trading relationships that can be initiated, reviewed and controlled using a powerful web and email based relationship management system.

The above-described system also provides a means of collecting, collating and recording consistent metrics that emanate from all inbound and outbound currents of user attention. This element enables marketers to facilitate detailed analysis and optimisation based on one consistent data source and subsequent reporting tool.

Finally, it should be appreciated that various modifications and/or additions may be made to the above-described method, system and components

without departing from the spirit or ambit of the present invention as defined in the claims appended hereto.

CLAIMS:

1. A method of tracking browsing activity of a user in a computer network including two or more independent computer network servers or server clusters, the user establishing a connection session with the computer network, the method including the steps of:
receiving a user request for content located on a first of the network servers or server clusters;
dynamically including a unique session identifier in content served from that first network server or server cluster to the user during the connection session;
transmitting the session identifier from the first to a second of the network servers or server clusters when a subsequent user request for content is made to the second network server or server cluster;
transmitting an event message including at least that session identifier to a central session manager upon the occurrence of a user browsing event during that connection session; and
logging session information derived from the event messages in a session data structure.

2. A method according to claim 1, wherein the event messages are generated and delivered by the user.

3. A method according to claim 1, wherein the event messages are generated and delivered by the network server providing content.

4. A method according to any one of the preceding claims, wherein the event messages are generated for session events, including any one or more of synchronisation event messages, impression events, result events, exposure events,

click events and/or person events.

5. A method of tracking upstream and downstream traffic flow between multiple sites within a computer network; the method including the steps of:
at the central session manager, receiving site entrance and exit data collected upon the transferral of user attention between the multiple sites;
identifying network site relationships from correlations in the site entrance and exit data; and
establishing a sequence of cause-and-effect relationships between the multiple sites.

6. A method according to claim 5, wherein the site entrance and exit data is captured by one or more tracking gates provided in one or more computer network servers hosting or forming part of a same network server cluster as the sites.

7. A method according to claim 5, wherein one or more of the tracking gates is an exit tracking gate that dynamically includes a site identifier and/or session identifier in the link to the second site.

8. A method according to claim 7, wherein the exit tracking gate includes a click sequence number or custom tracking data.

9. A method according to any one of claims 6 to 8, wherein one or more of the tracking gates is an entrance tracking gate that attempts to identify and record the site identifier and/or session identifier from data encoded in the user request.

10. A method according to claim 9, wherein the data encoded in the user request

includes any one or more of a URL, a cookie and/or a referrer.

11. A method according to any one of claims 6 to 10, wherein one or more of the tracking gates is a clickthrough tracking gate that records both first and second site identifier at the first site.

12. A method according to claim 11, wherein the clickthrough gate identifies or creates the session identifier, click sequence number and custom tracking data from the first site.

13. A method according to any one of claims 5 to 12, wherein the site entrance and exit data are captured by session management based techniques.

14. A method according to claim 13, wherein session management based techniques include analysing session data at the second site to identify entrance points from the first site.

15. A method according to either one of claims 13 or 14, wherein the session management based techniques include identifying external events directly resulting from an external hyperlink, and analysing external events to identify the first site.

16. A method according to claim 15, wherein the external events are analysed by extracting a session identifier, a click sequence number and/or custom tracking data from the first site.

17. A method according to either one of claims 15 or 16, wherein the external events are analysed by using any one or more of a flagged entrance algorithm, a sequencing based algorithm, or an internal identifier algorithm.

18. A method according to any one of claims 13 to 17, wherein session management based techniques include applying a clicktrees tree index structure to identify session exit points.

19. A method according to any one of claims 5 to 18, wherein the site entrance and exit data is captured by session-less entrance detection means.

20. A method according to claim 19, wherein the session-less entrance detection means uses an entrance recorded flag.

21. A method according to either one of claims 19 or 20, wherein the session-less entrance detection means uses HTTP Host and Referrer Request Header Analysis.

22. A method according to any one of claims 5 to 21, wherein the site entrance and exit data are captured by generating transition identifiers.

23. A method according to any one of claims 5 to 22, wherein the sequence of cause-and-effect relationships is established by constructing a time-based clickstream tree index structure of use attention transfer.

24. A method of maintaining connection session state in a computer network including multiple sites, the user establishing a connection session with the computer network, information characterising the user being captured whenever the user interacts with each site to generate a user browsing signature; the method including the steps of:

at a central session manager, receiving the user browsing signature; and

deriving session information for the connection session from the transmitted

browsing signatures.

25. A method according to claim 24, wherein the captured information includes information identifying a network presence of the user.

26. A method according to claim 25, wherein information identifying a network presence of the user includes the user's network address.

27. A method according to claim 26, wherein the computer network is the Internet, and the user's network address is an IP address.

28. A method according to any one of claims 24 to 27, wherein the captured information also includes an data packet header.

29. A method according to claim 28, wherein the data packet header is an HTTP header.

30. A method according to either one of claims 28 or 29, wherein the data packet header is a referrer request header, an accept language request header or a user agent request header.

31. A method according to any one of claims 24 to 30, wherein the captured information includes user custom data.

32. A method according to any one of claims 24 to 31, wherein a source data string is constructed from the captured information.

33. A method according to claim 32, wherein a hashing function is applied to the

source data string.

34. A method of tracking browsing activity of a user in a computer network including multiple sites, the user establishing a connection session with the computer network, impression events being generated when content is requested by the user from a network site, and result events being generated when the user completes a step towards a connection session result; the method including the step of:

at a central session manager, receiving messages representative of the impression and results events; and

interleaving impression events and result events to derive result progress path information for the user.

35. A method according to claim 34, wherein the results events include at least result start, result report and result end events.

36. A central session manager including a processing means and associated memory means, including code storage means for storing computer program code to cause the central sessions manager to carry out a method according to any one of claims 1 to 35.

37. Computer program code for storage in memory means associated with processing means, the processing means and memory means forming part of a central session manager, the computer program code causing the central session manager to carry out a method according to any one of claims 1 to 36.

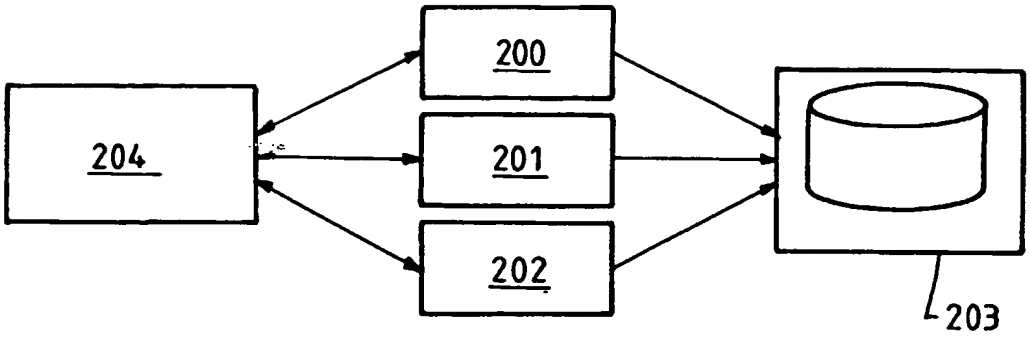


FIG. 1.

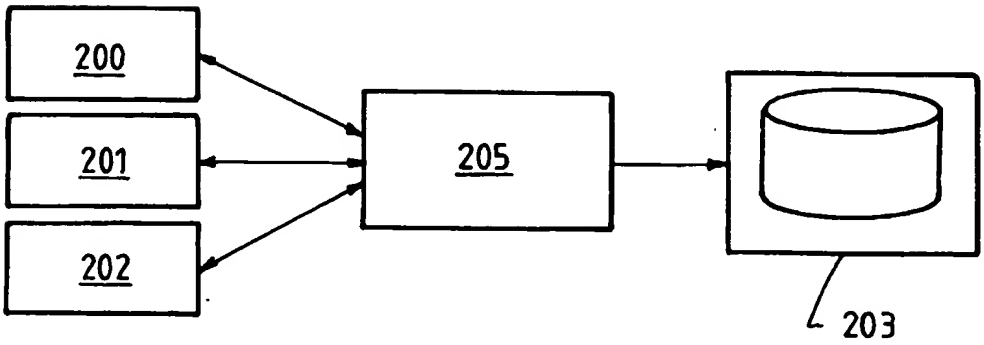
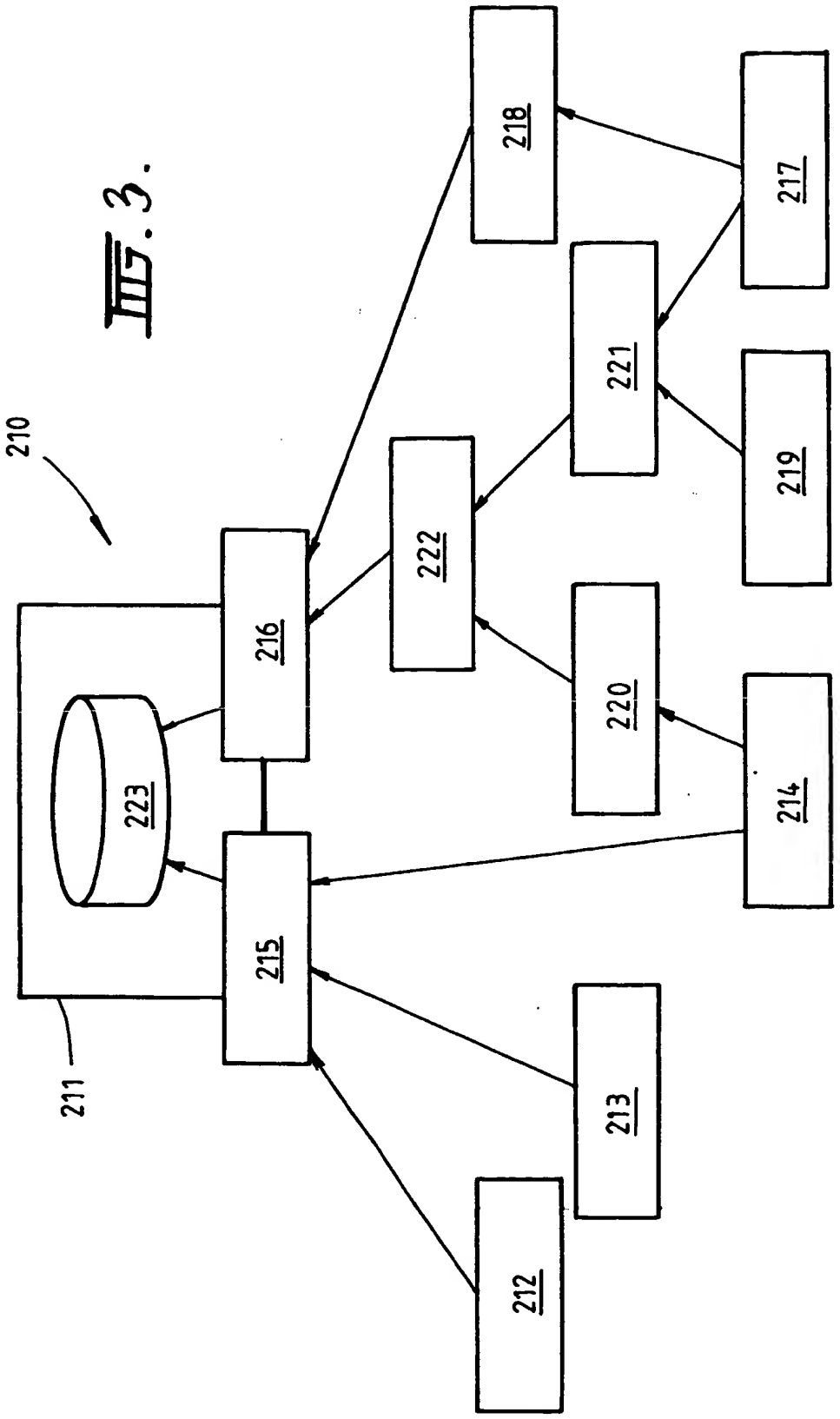


FIG. 2.

Fig. 3.



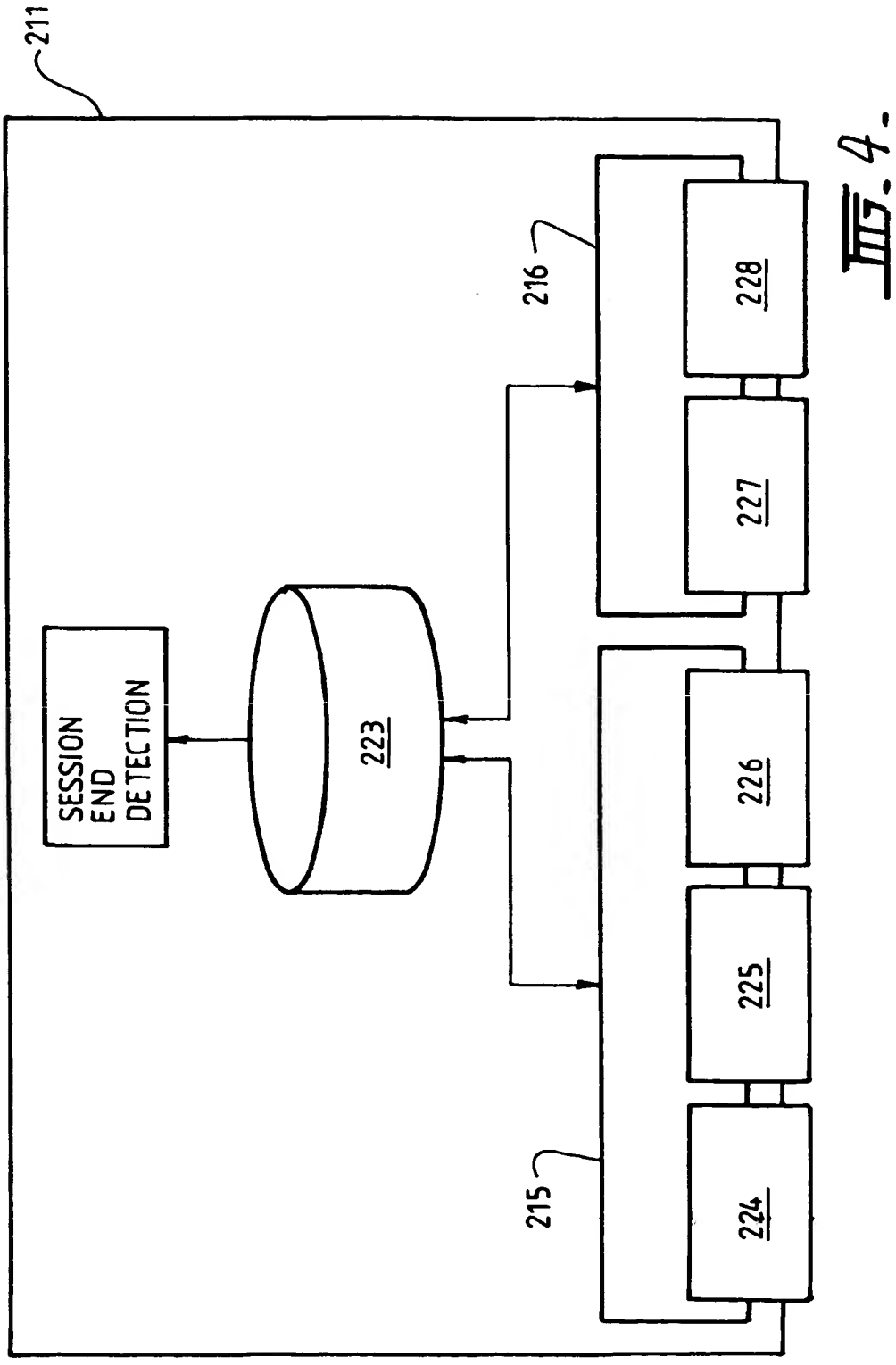


FIG. 4.

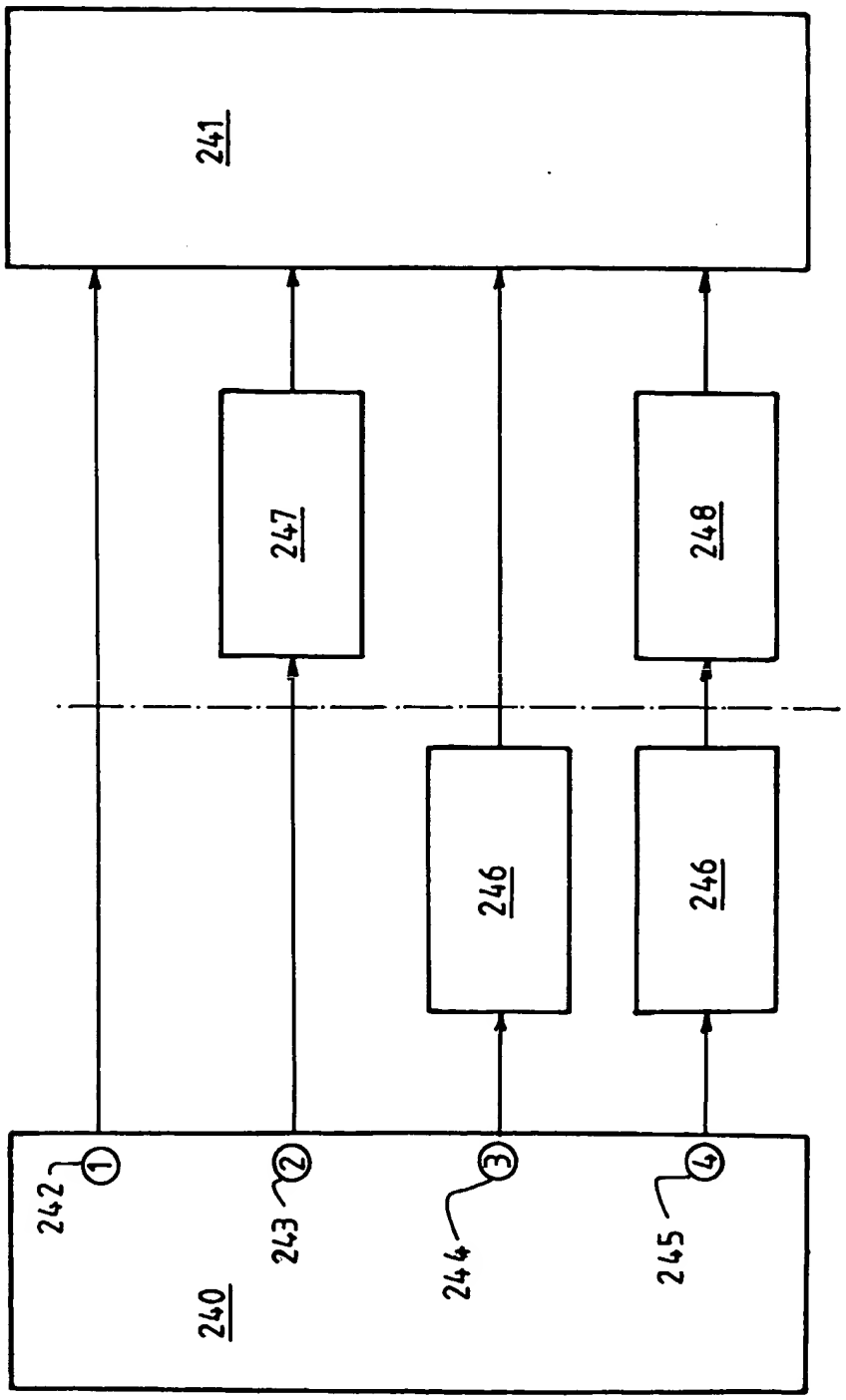


Fig. 5.

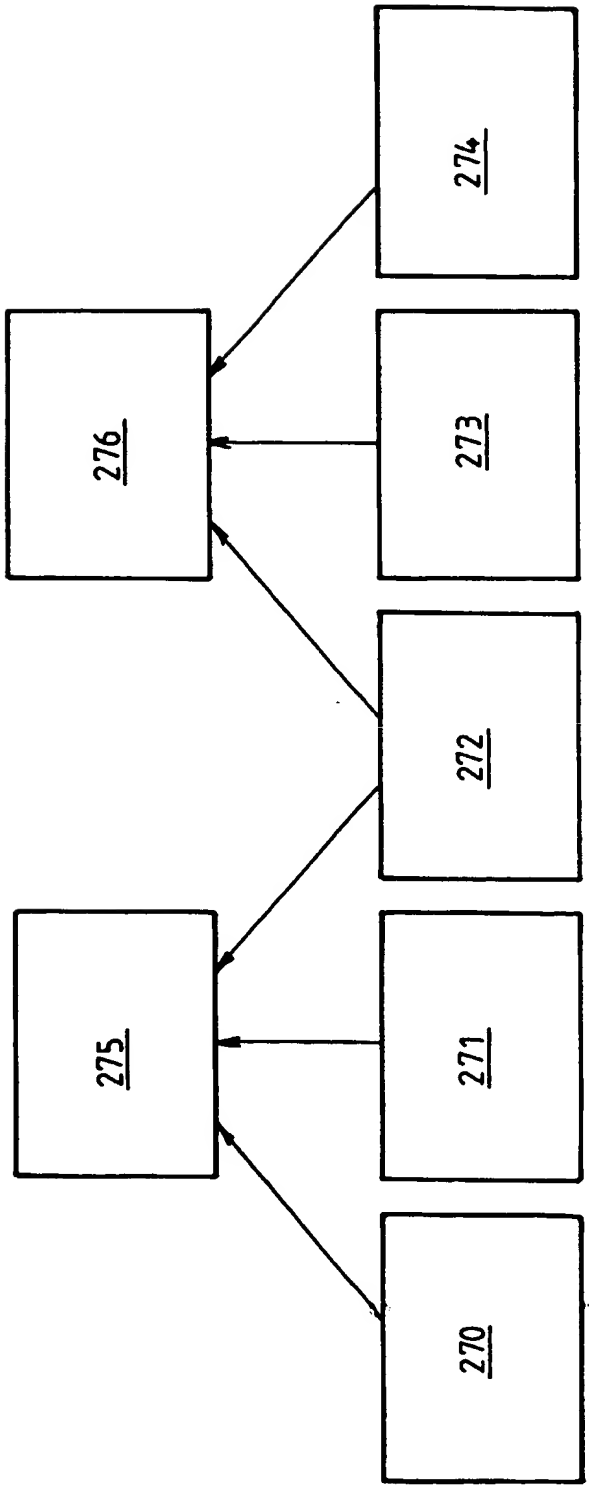


Fig. 6.

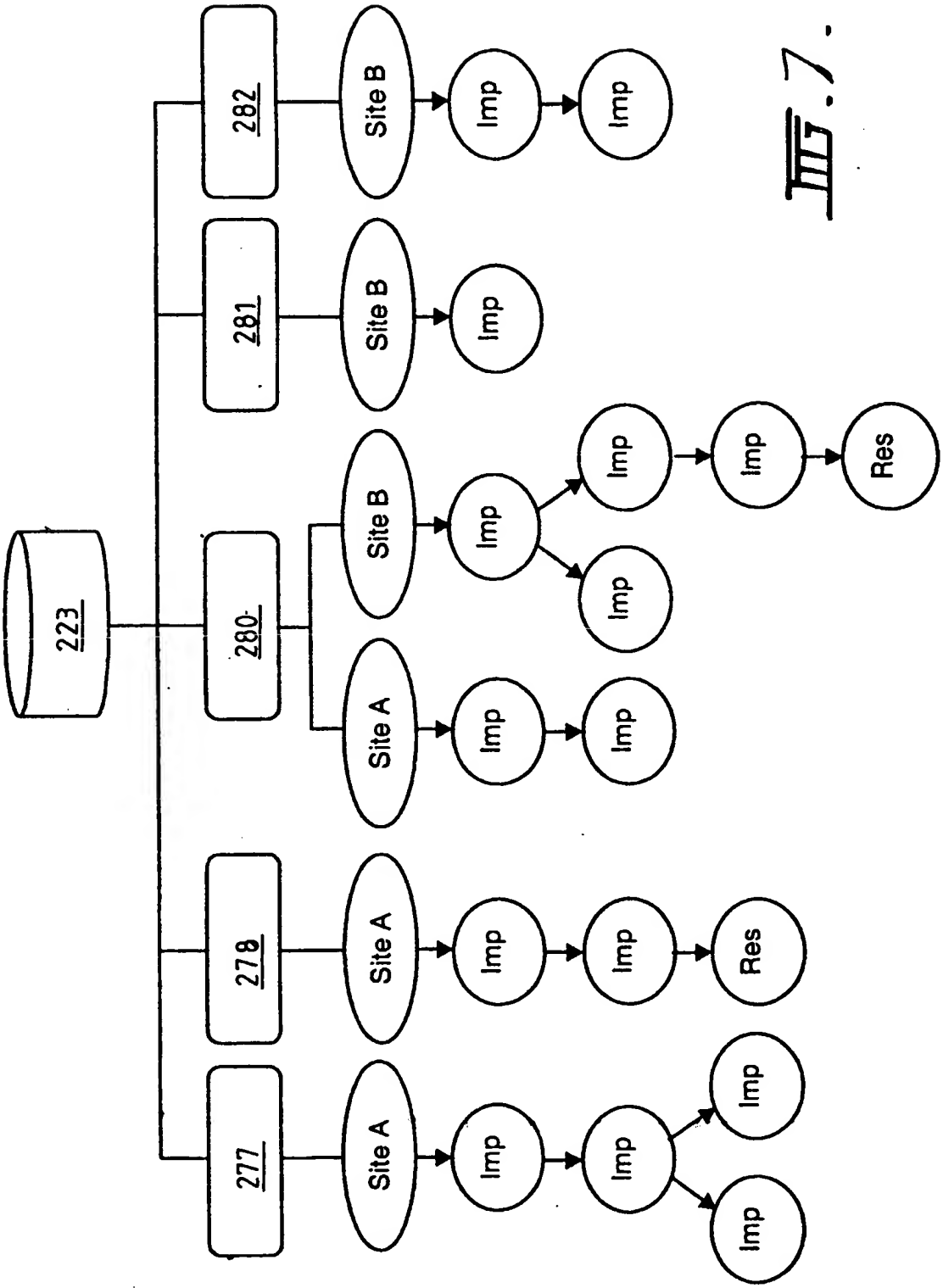


FIG. 7.

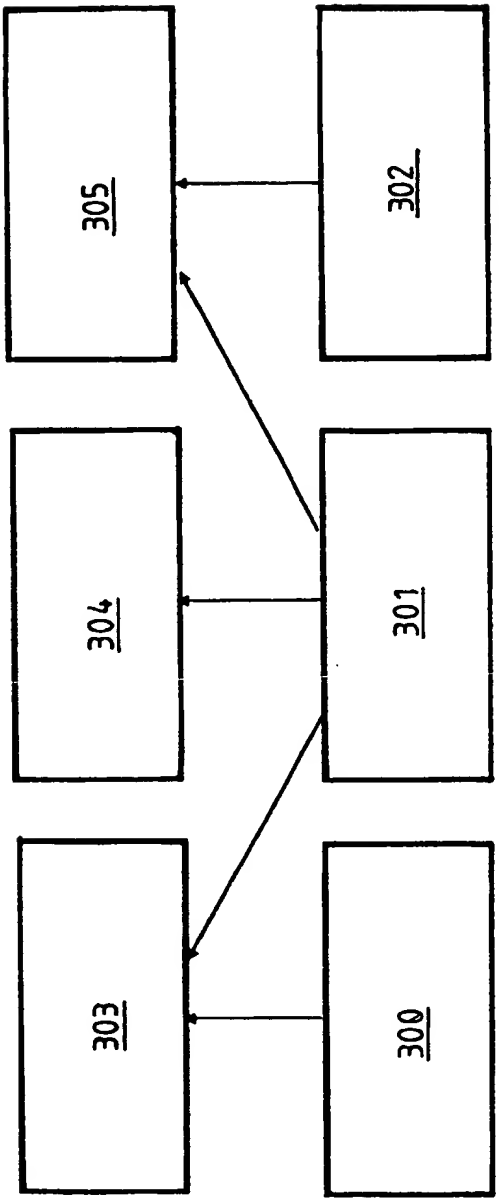


Fig. 8.

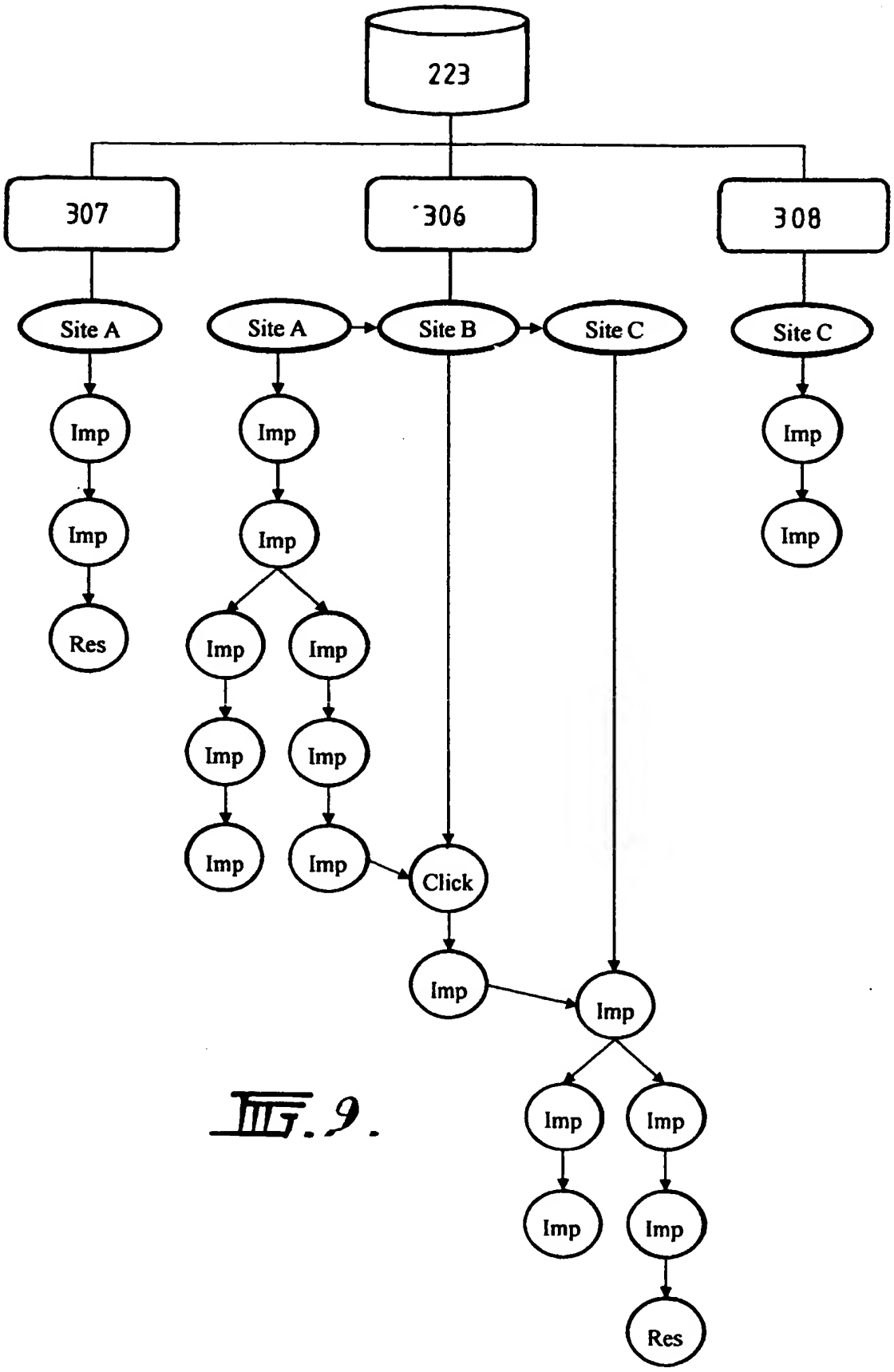

223
307
306
308
Site A
Site A
Site B
Site C
Site C
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Imp
Res
Click
Res

FIG. 9.

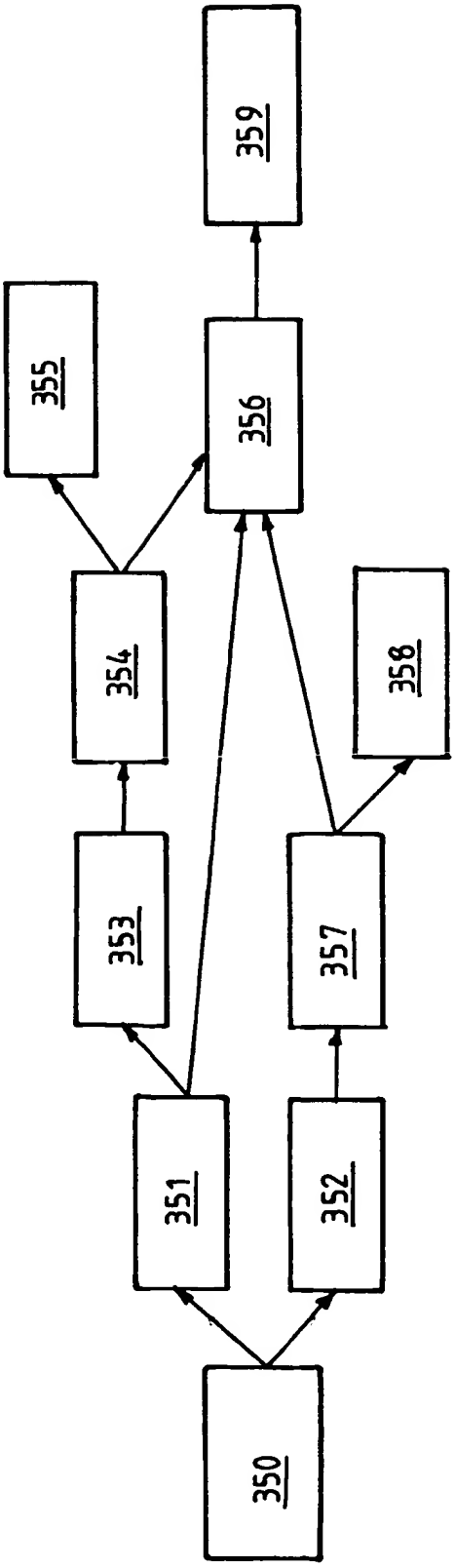
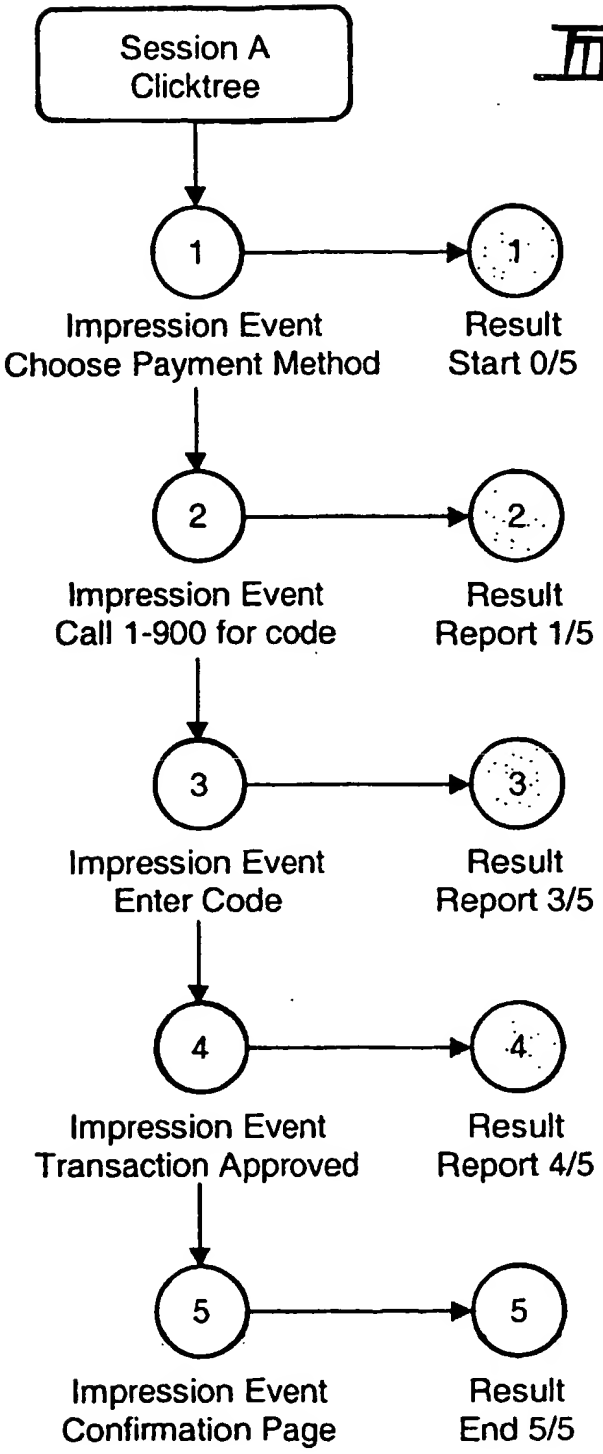


Fig. 10.

FIG. 11.


Session A
Clicktree
1
1
Impression Event
Choose Payment Method
Result
Start 0/5
2
2
Impression Event
Call 1-900 for code
Result
Report 1/5
3
3
Impression Event
Enter Code
Result
Report 3/5
4
4
Impression Event
Transaction Approved
Result
Report 4/5
5
5
Impression Event
Confirmation Page
Result
End 5/5

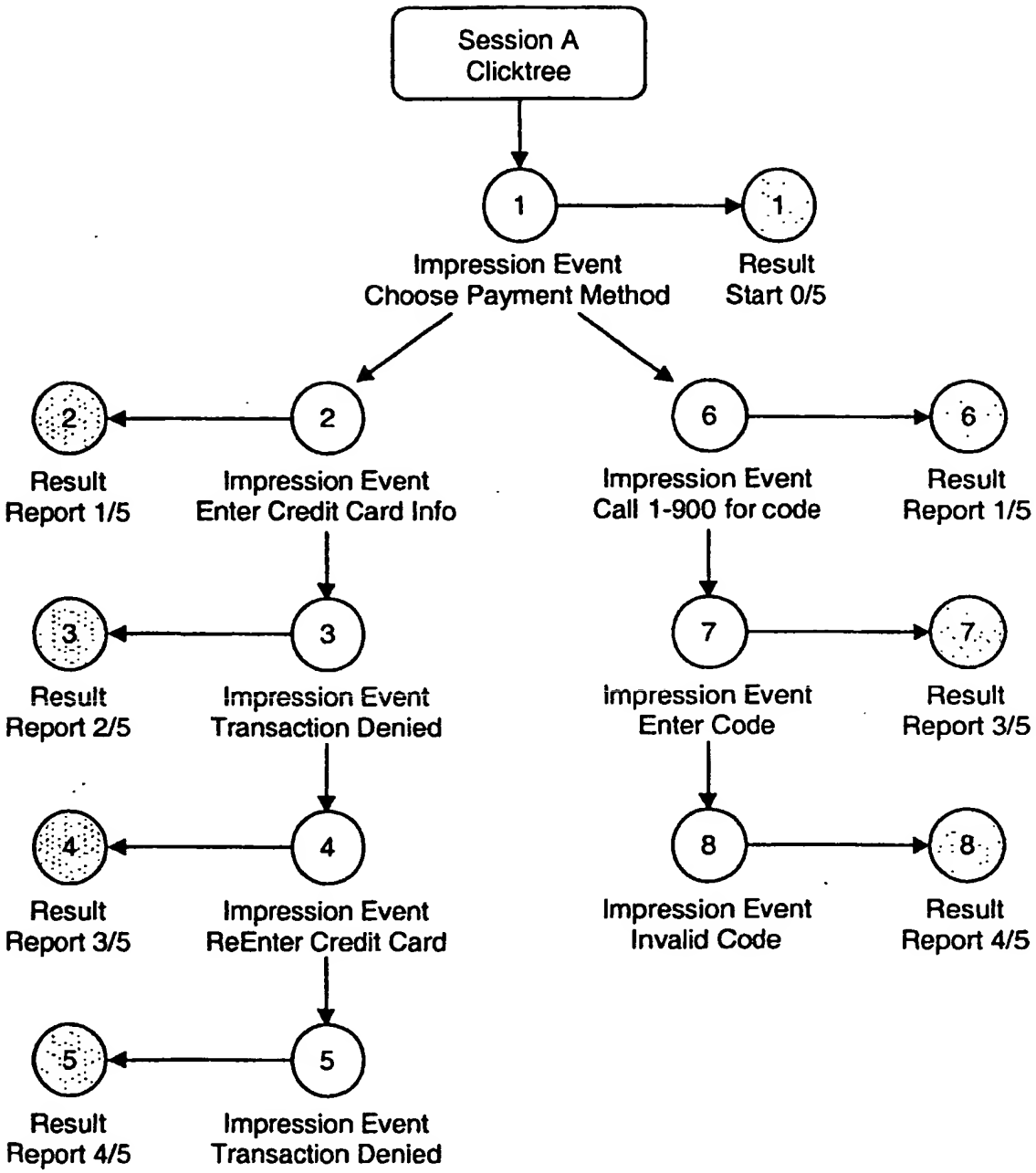

Session A
Clicktree
1
Impression Event
Choose Payment Method
1
Result
Start 0/5
2
Result
Report 1/5
2
Impression Event
Enter Credit Card Info
6
Impression Event
Call 1-900 for code
6
Result
Report 1/5
3
Result
Report 2/5
3
Impression Event
Transaction Denied
7
Impression Event
Enter Code
7
Result
Report 3/5
4
Result
Report 3/5
4
Impression Event
ReEnter Credit Card
8
Impression Event
Invalid Code
8
Result
Report 4/5
5
Result
Report 4/5
5
Impression Event
Transaction Denied

FIG. 12

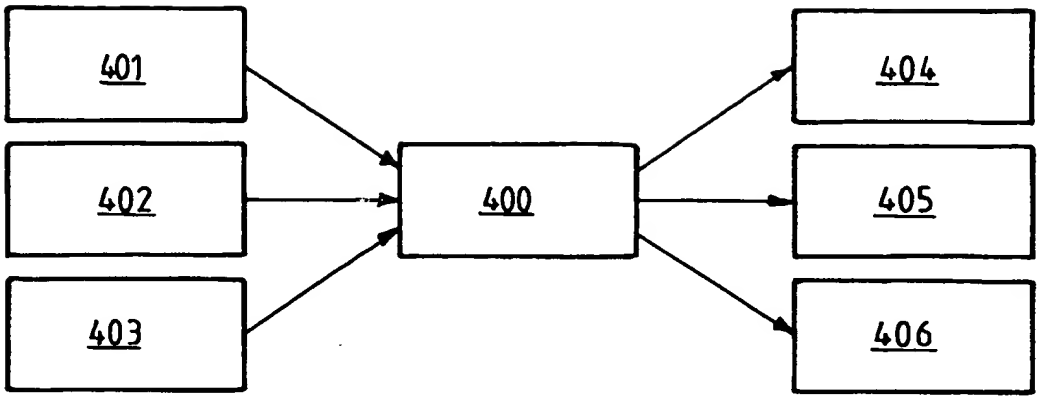


FIG. 13.

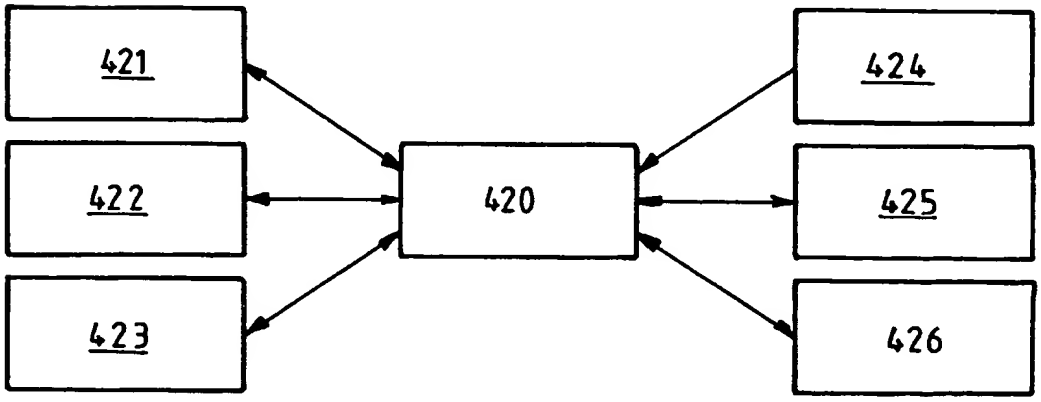


FIG. 14.

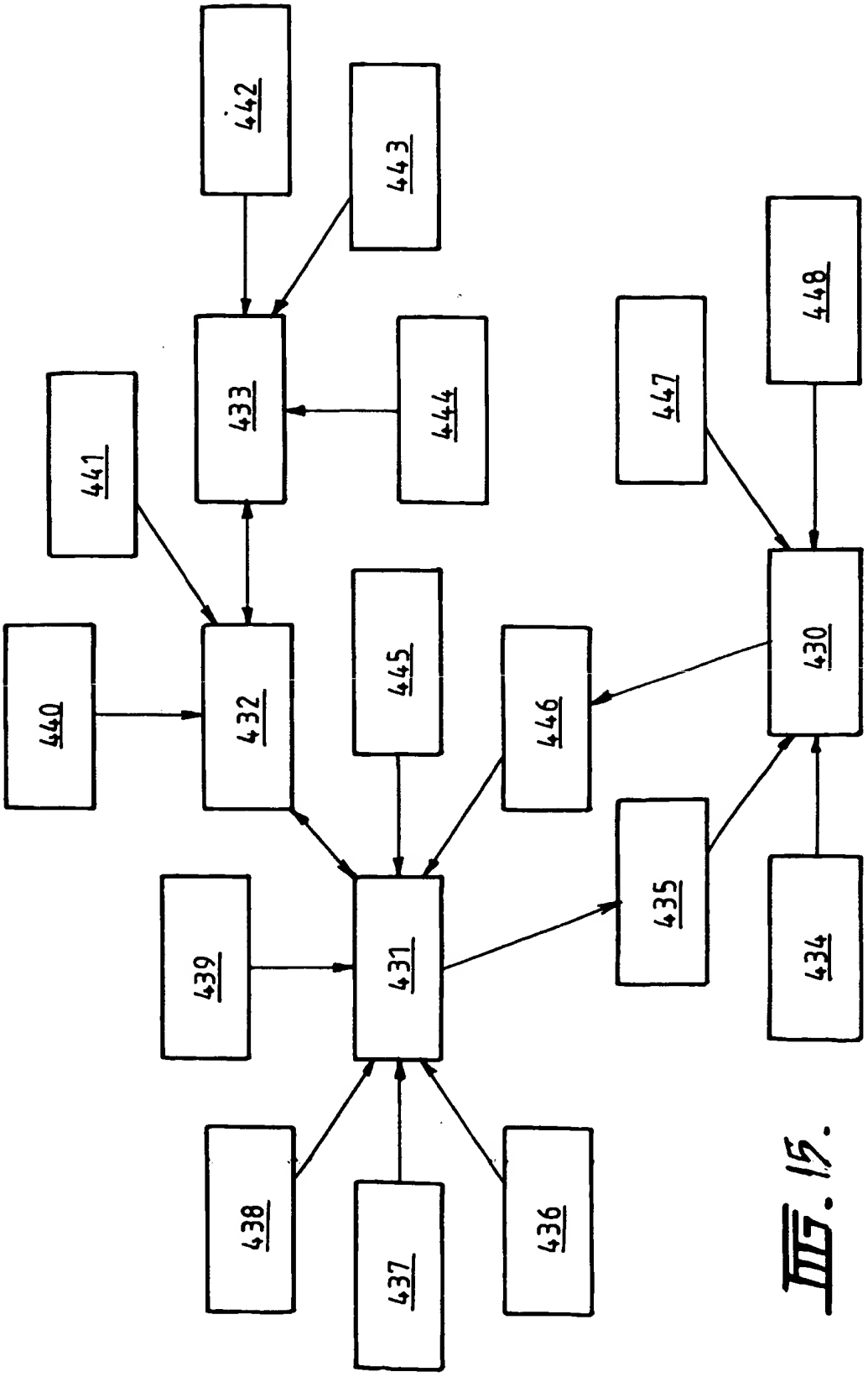

442
443
433
444
441
440
432
445
446
430
447
448
435
434
439
431
438
437
436

FIG. 15.


500
501
502
503
504

FIG. 16.

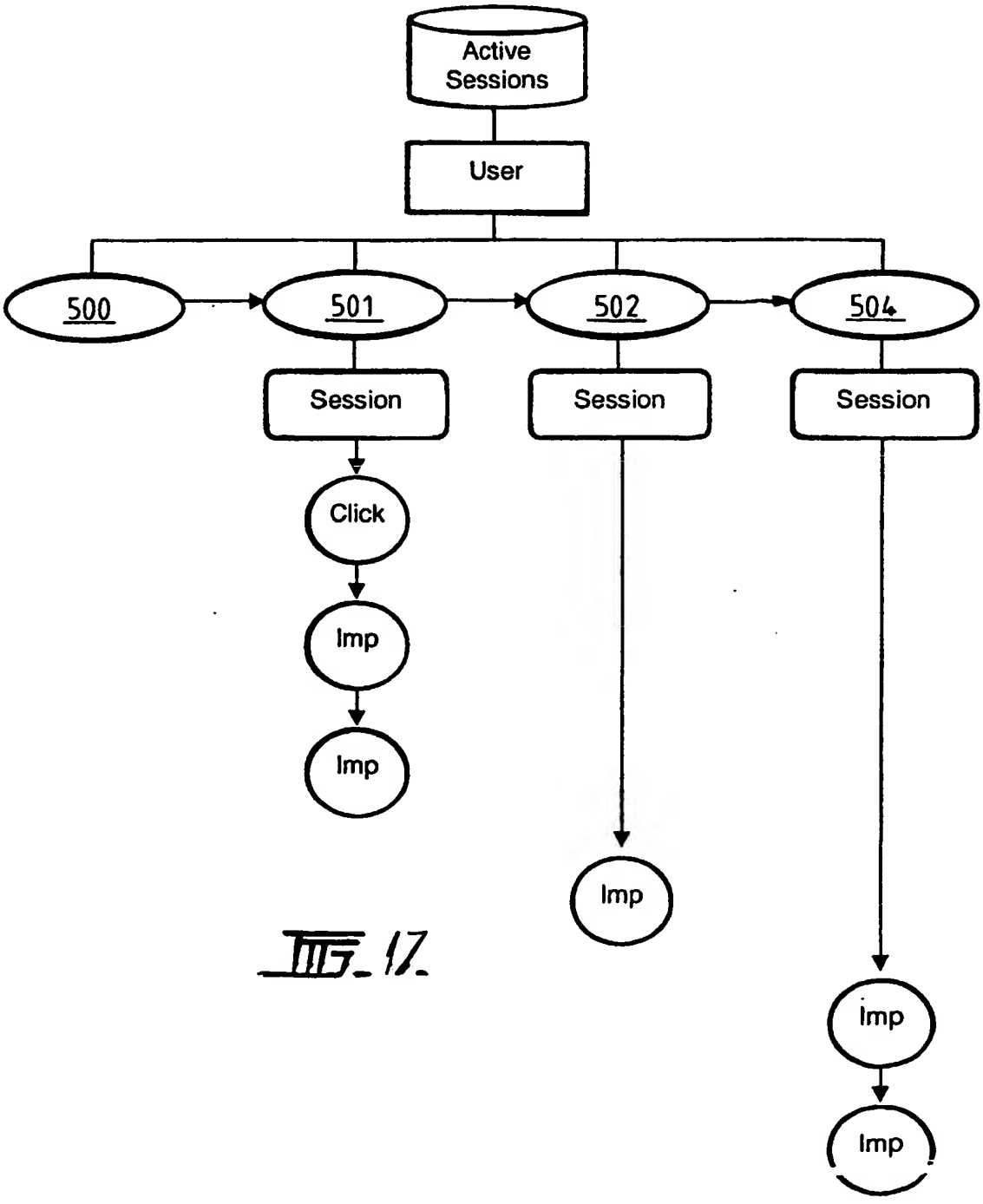

Active Sessions
User
500
501
502
504
Session
Session
Session
Click
Imp
Imp
Imp
Imp
Imp

FIG. 17.

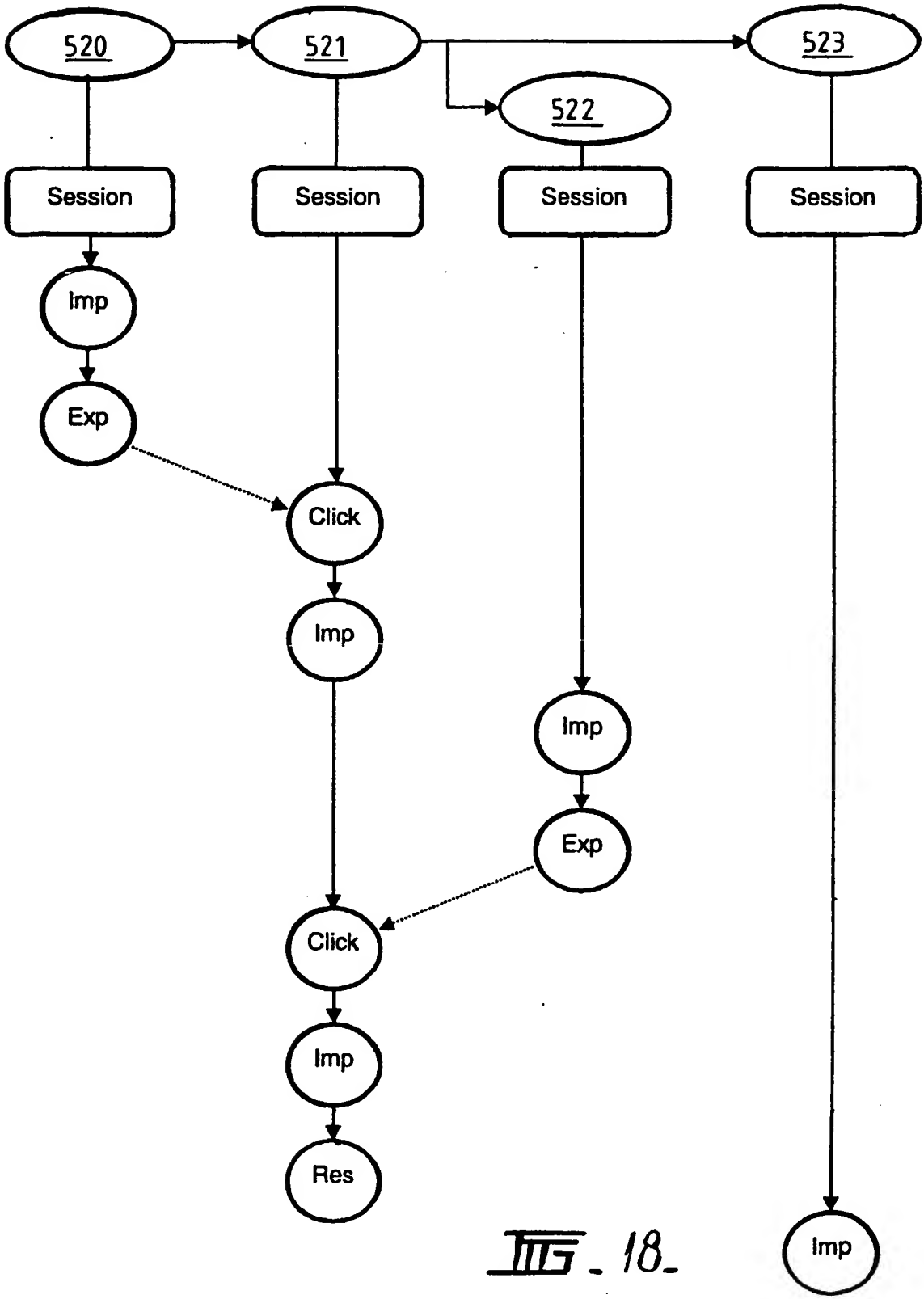

520
521
523
522
Session
Session
Session
Session
Imp
Exp
Click
Imp
Imp
Exp
Click
Imp
Res
Imp

FIG. 18.

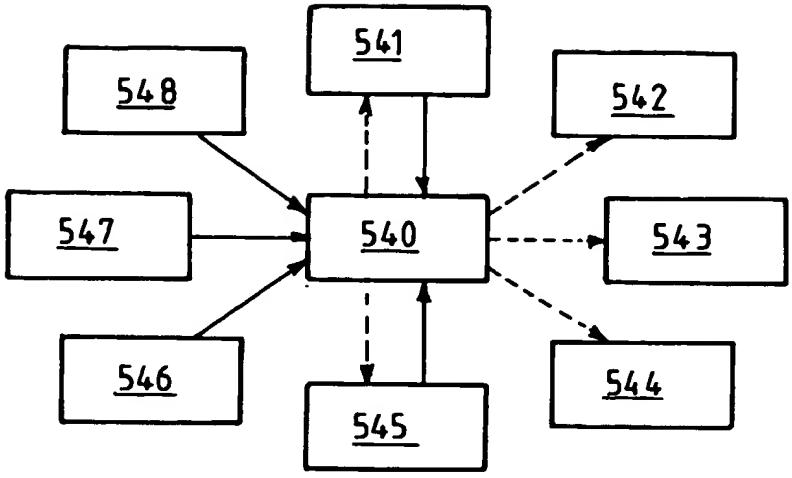


Fig. 19.

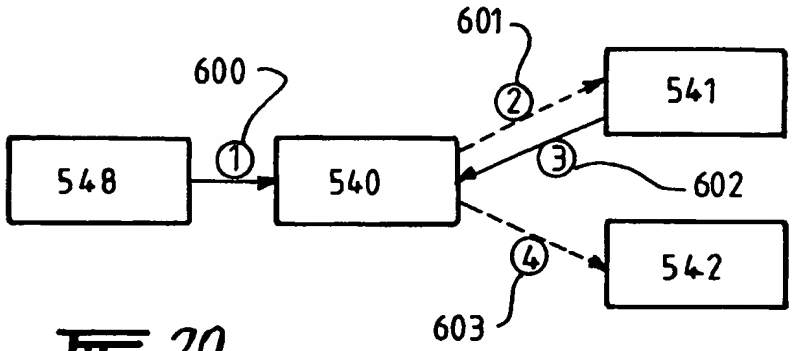


Fig. 20.

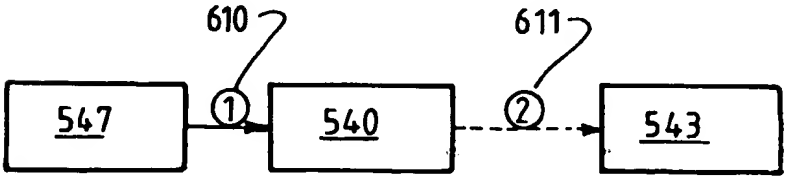


Fig. 21.

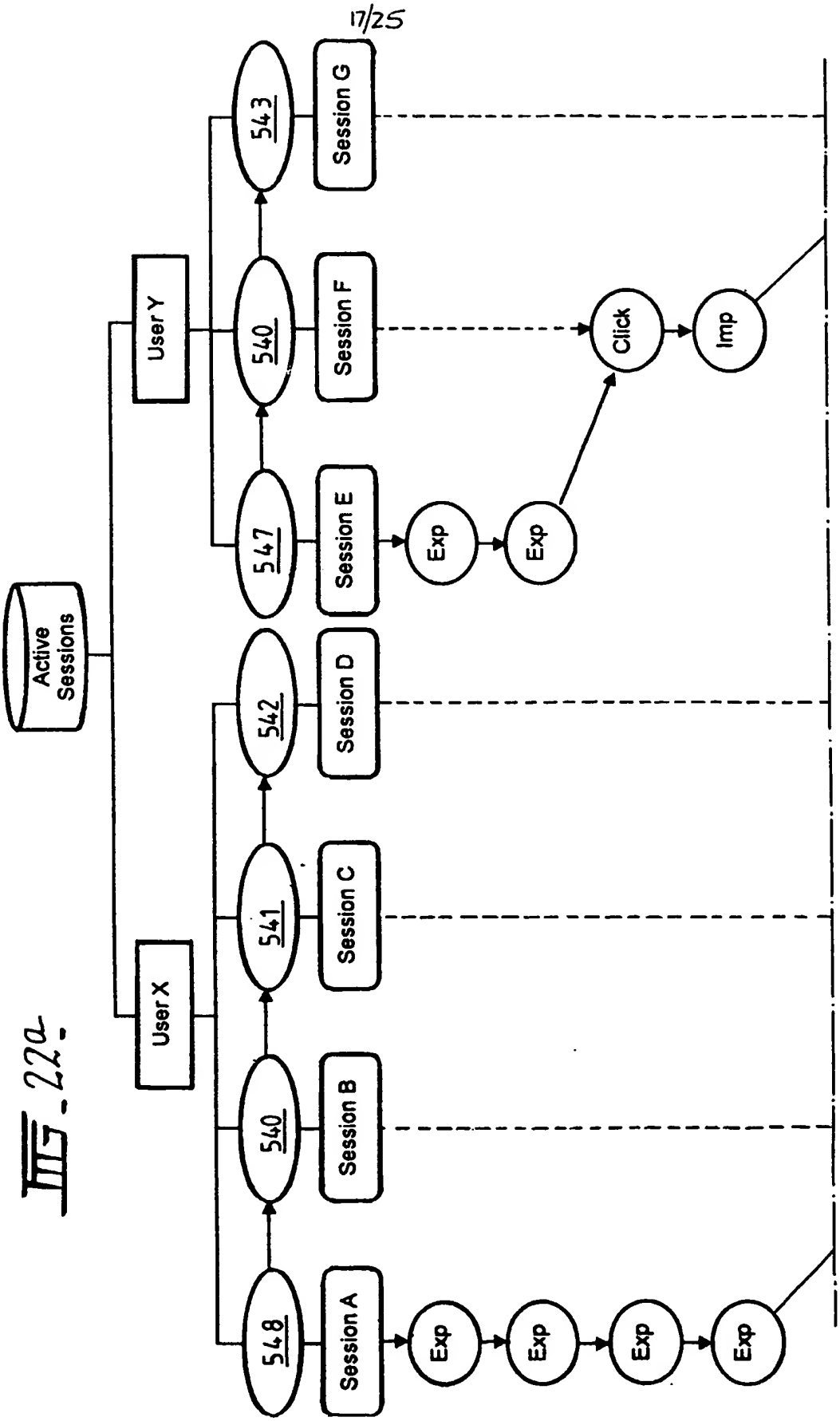

Active Sessions
User X
User Y
548
540
541
542
547
540
543
Session A
Session B
Session C
Session D
Session E
Session F
Session G
Exp
Exp
Exp
Exp
Exp
Exp
Click
Imp

FIG. 22a

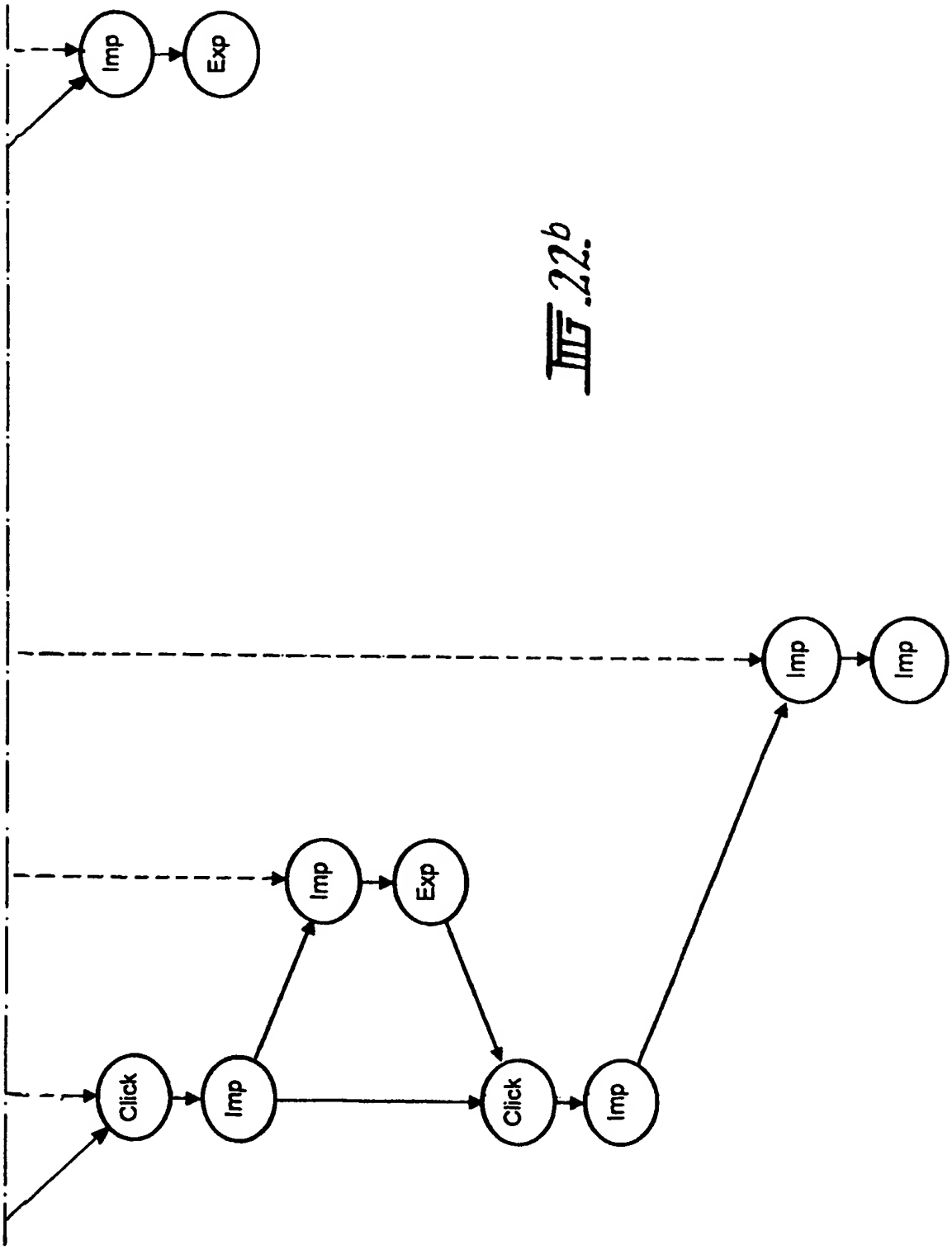


Fig. 22b

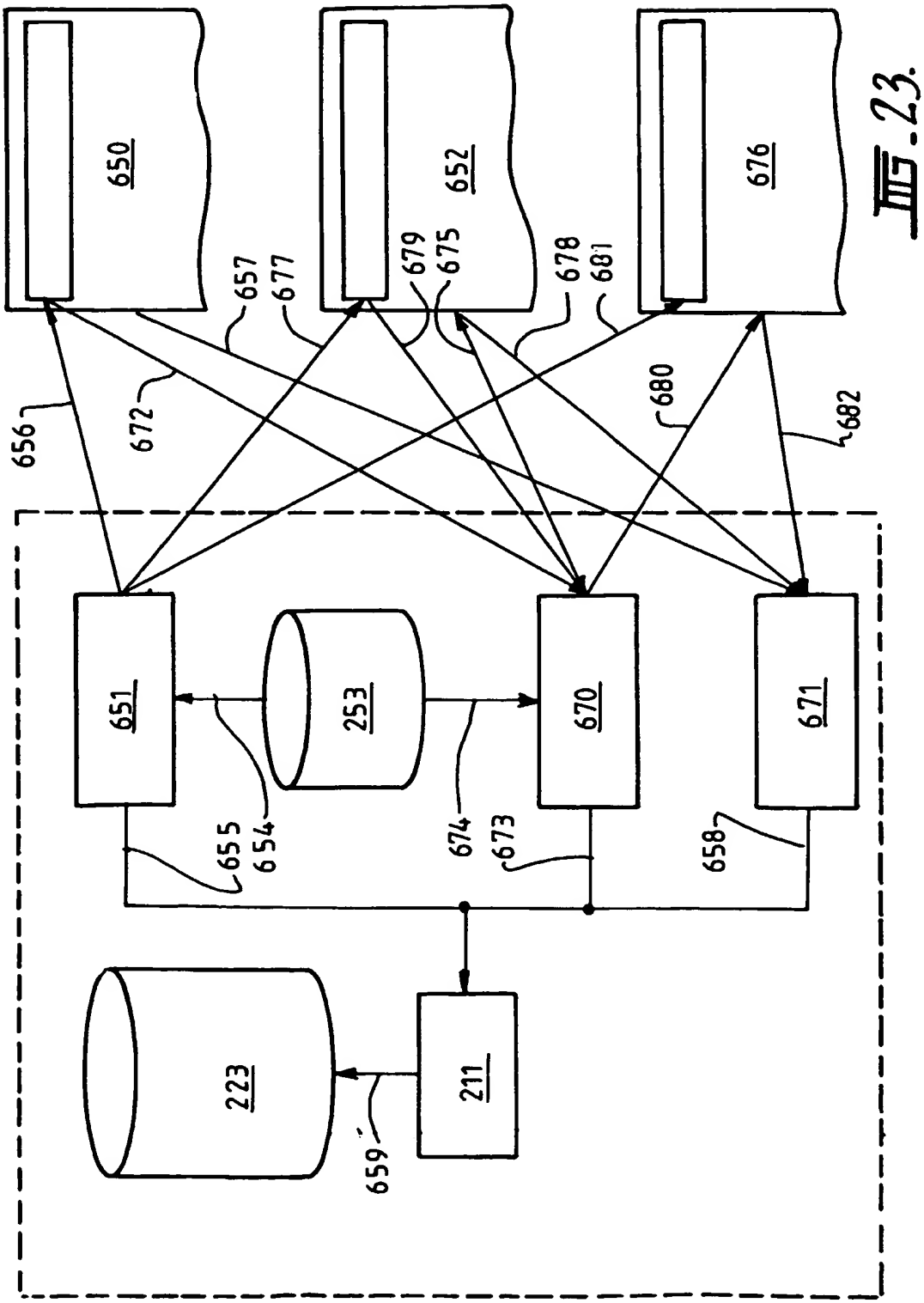

650
652
676
657
677
679
675
678
681
656
672
680
682
651
253
670
671
655
654
674
673
658
223
211
659

FIG. 23.

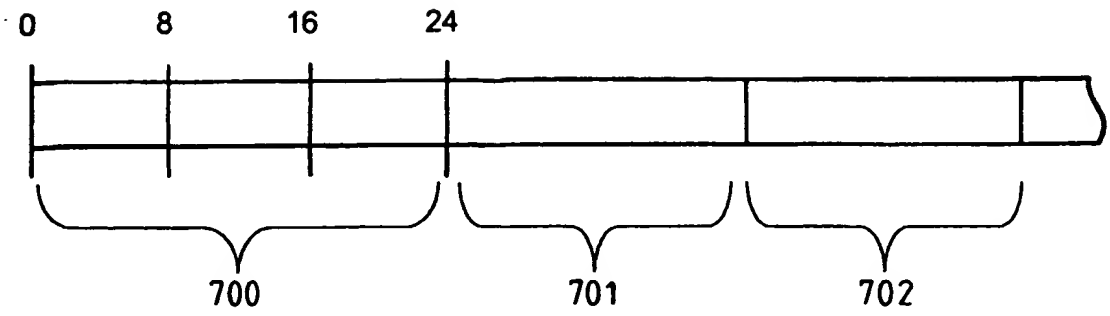

0
8
16
24
700
701
702

FIG. 24.

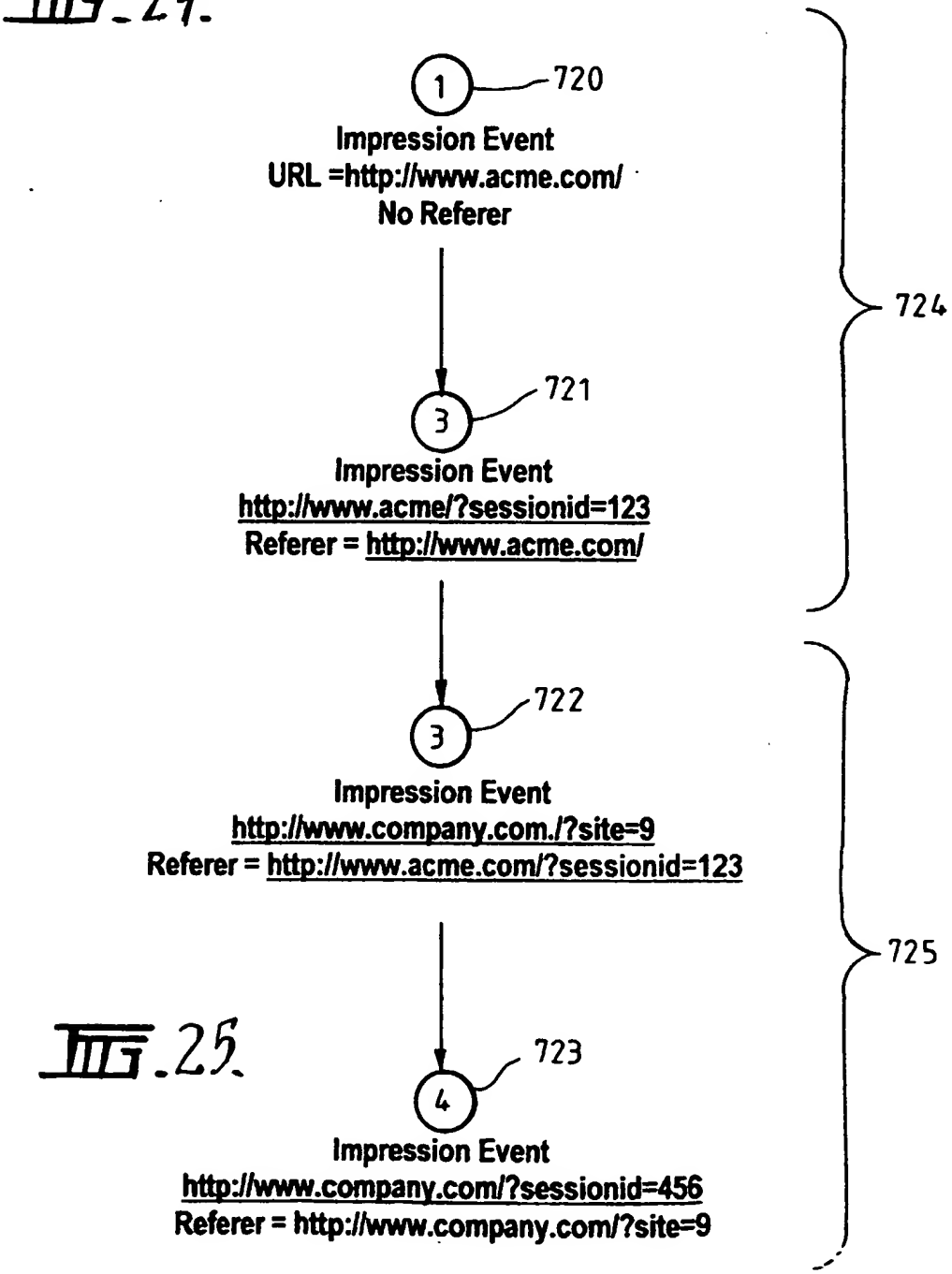

1
720
Impression Event
URL =http://www.acme.com/
No Referer
724
3
721
Impression Event
http://www.acme/?sessionid=123
Referer = http://www.acme.com/
3
722
Impression Event
http://www.company.com./?site=9
Referer = http://www.acme.com/?sessionid=123
725
4
723
Impression Event
http://www.company.com/?sessionid=456
Referer = http://www.company.com/?site=9

FIG. 25.

FIG. 26a

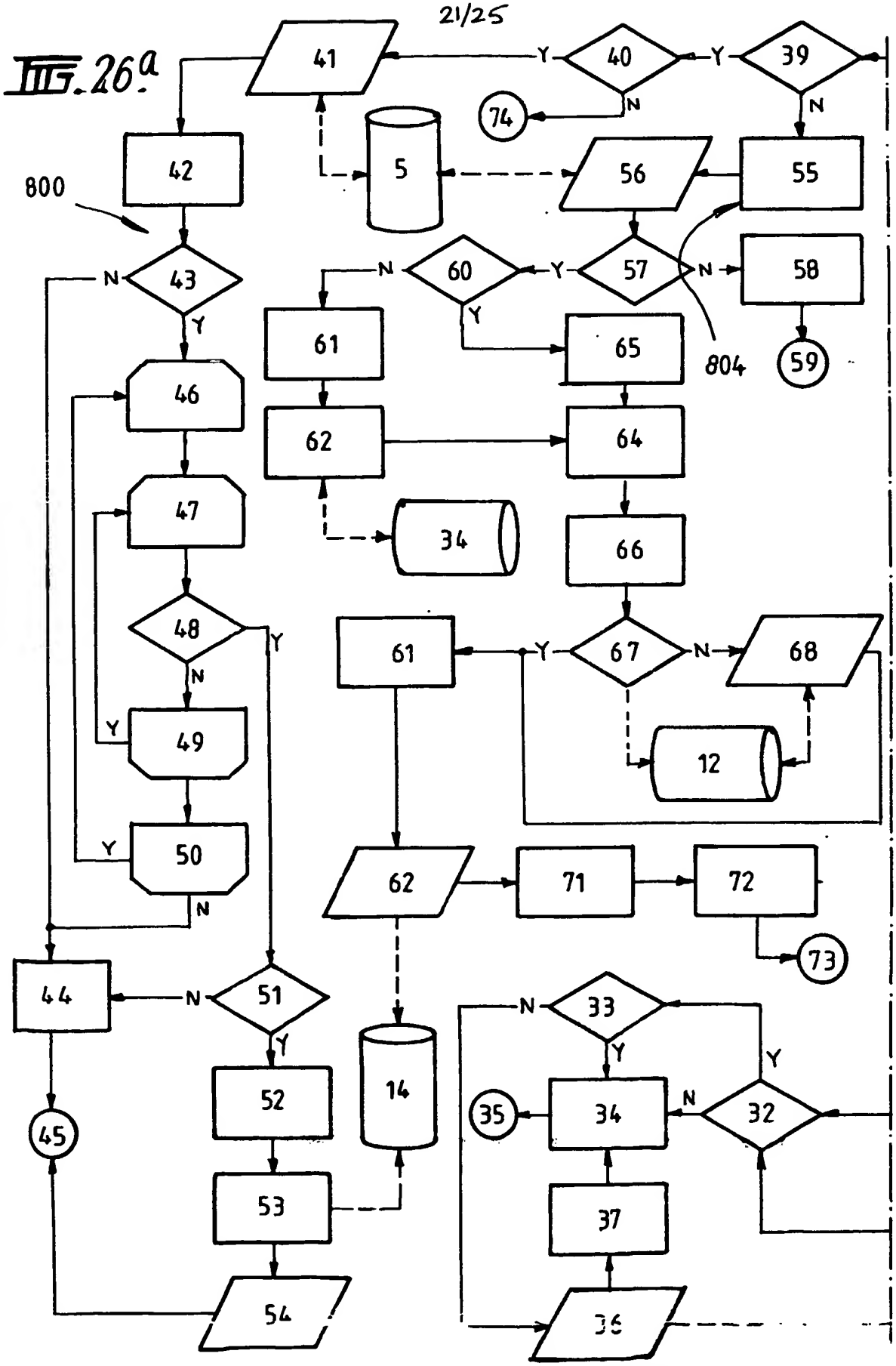
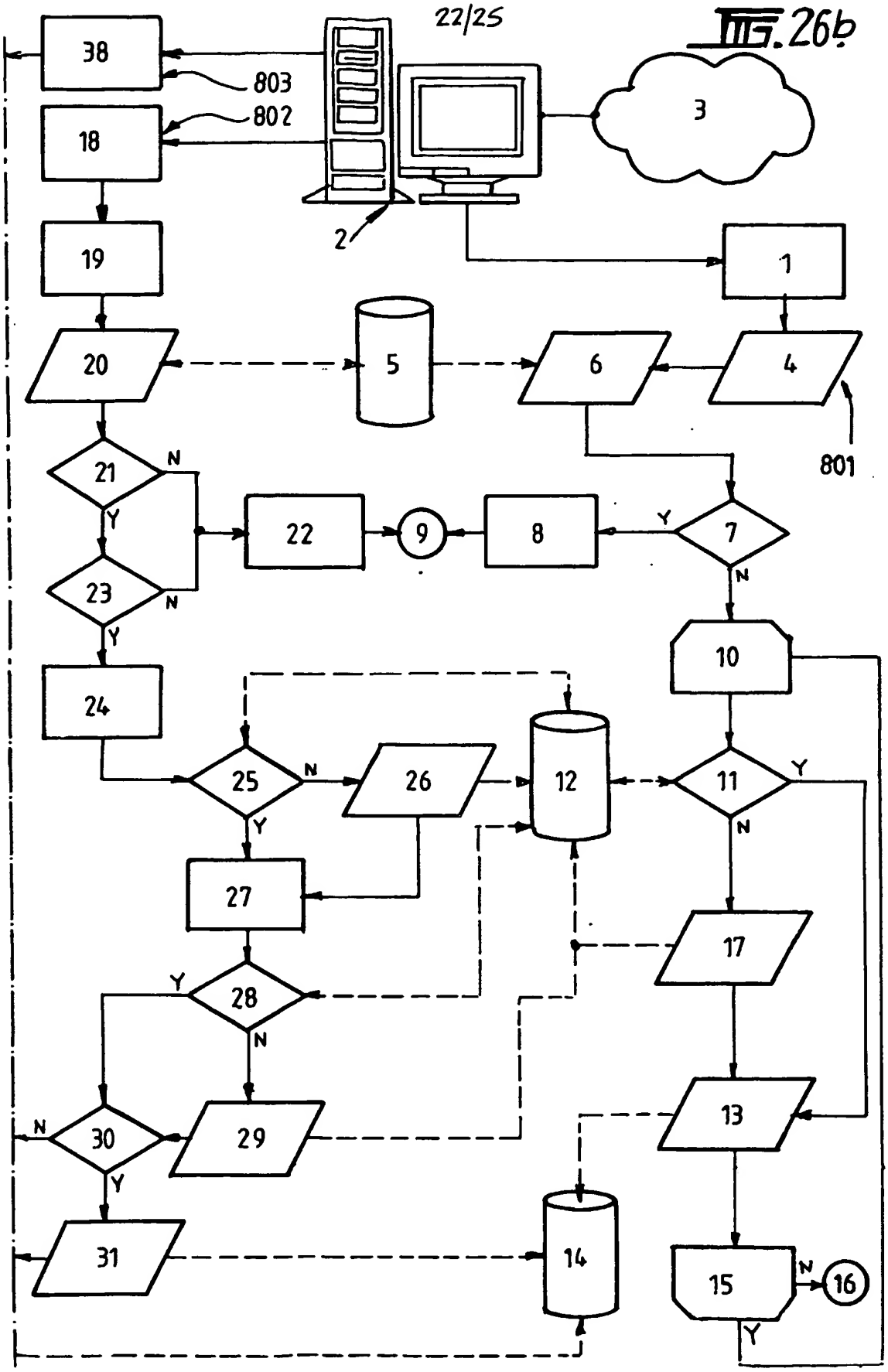

41
40
39
74
42
5
56
55
800
43
60
57
58
61
65
804
59
46
62
64
47
34
66
48
61
67
68
49
12
50
62
71
72
73
44
51
52
14
33
45
53
34
32
35
37
54
36

FIG. 26b


38
803
802
18
3
2
19
1
20
5
6
4
801
21
N
Y
22
9
8
Y
7
N
23
N
Y
10
24
25
N
26
12
11
Y
Y
N
27
17
28
Y
N
30
N
29
13
Y
31
14
15
N
16
Y

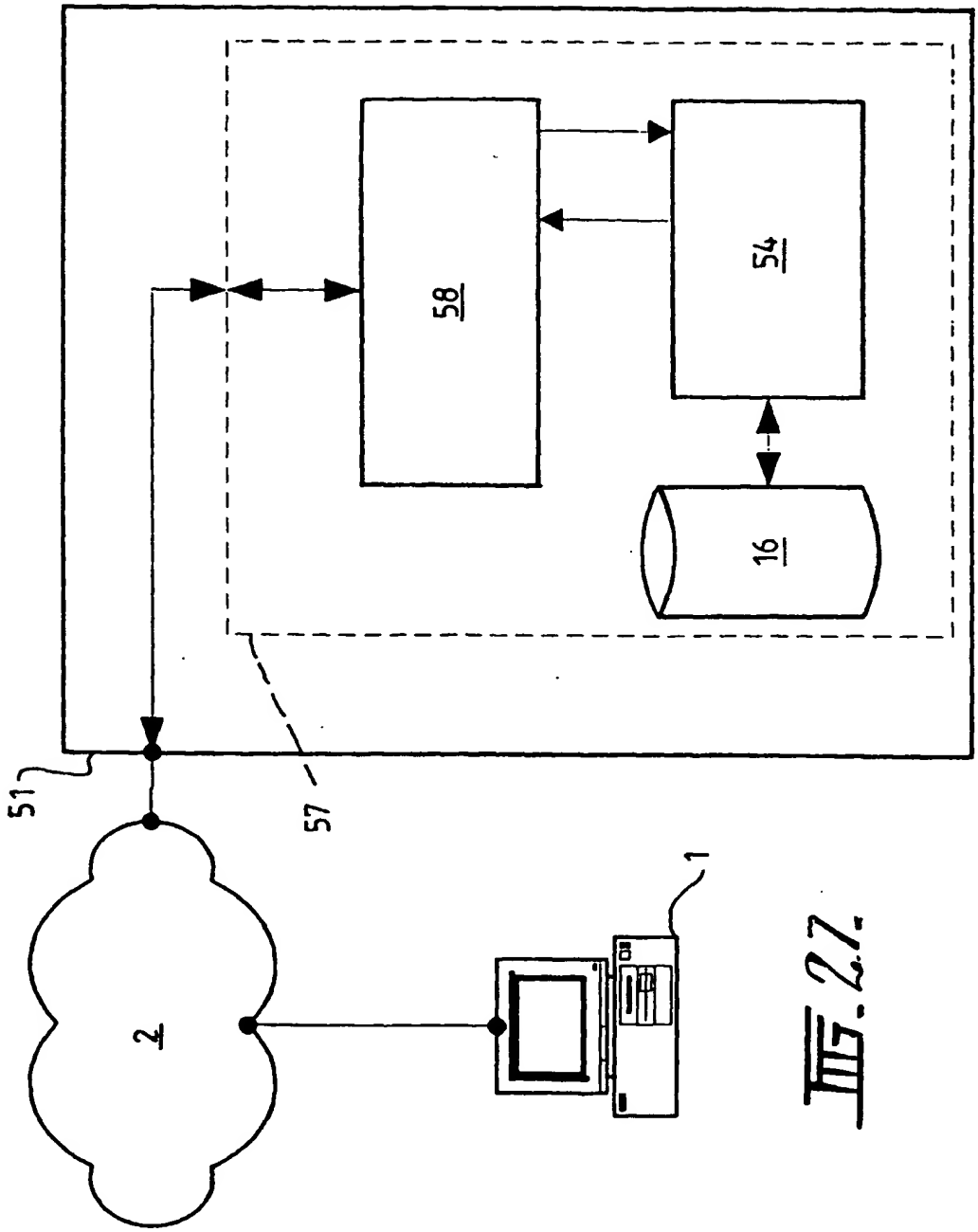


FIG. 27.

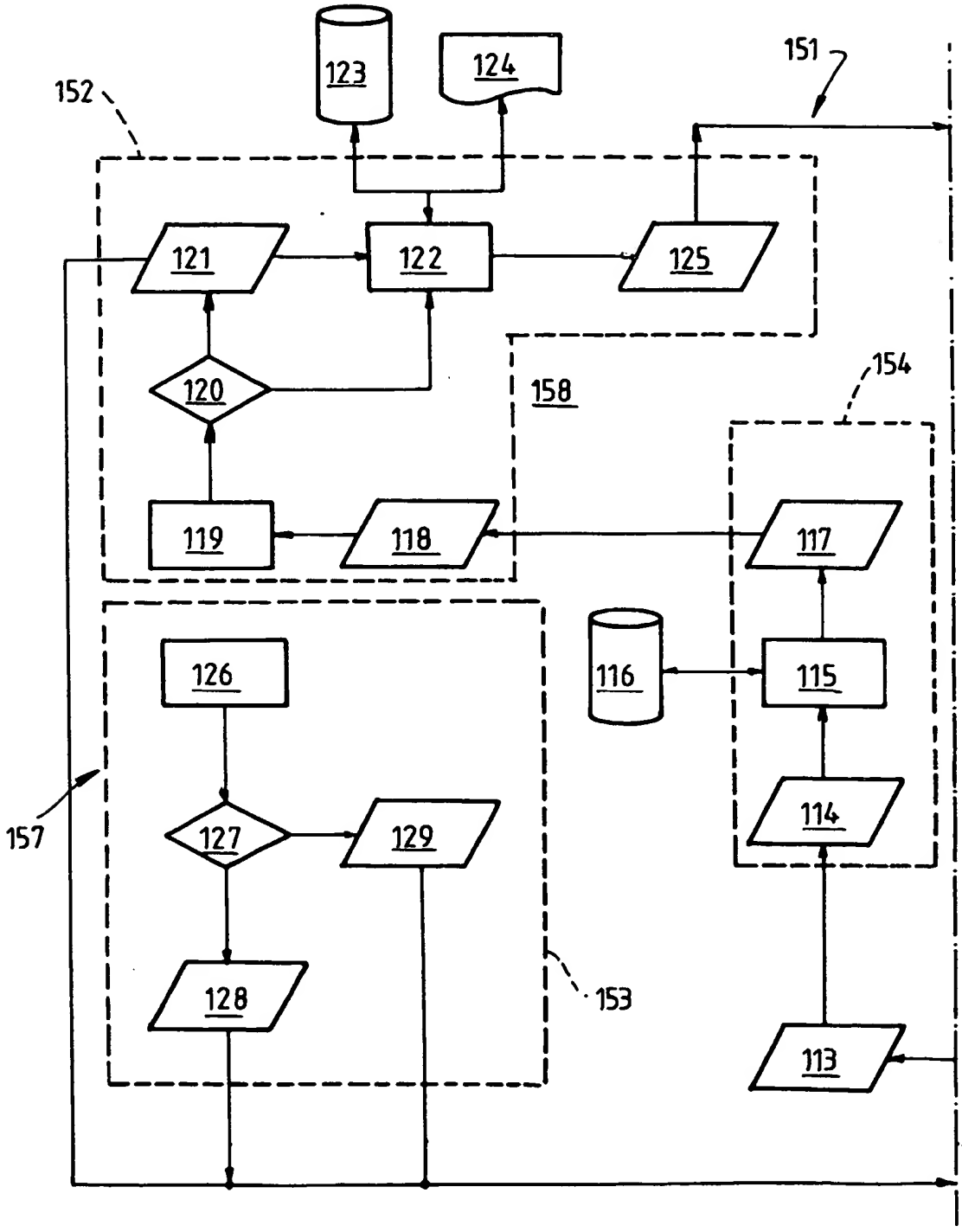

123
124
151
152
121
122
125
120
158
154
119
118
117
116
115
126
114
157
127
129
128
153
113

FIG. 28a

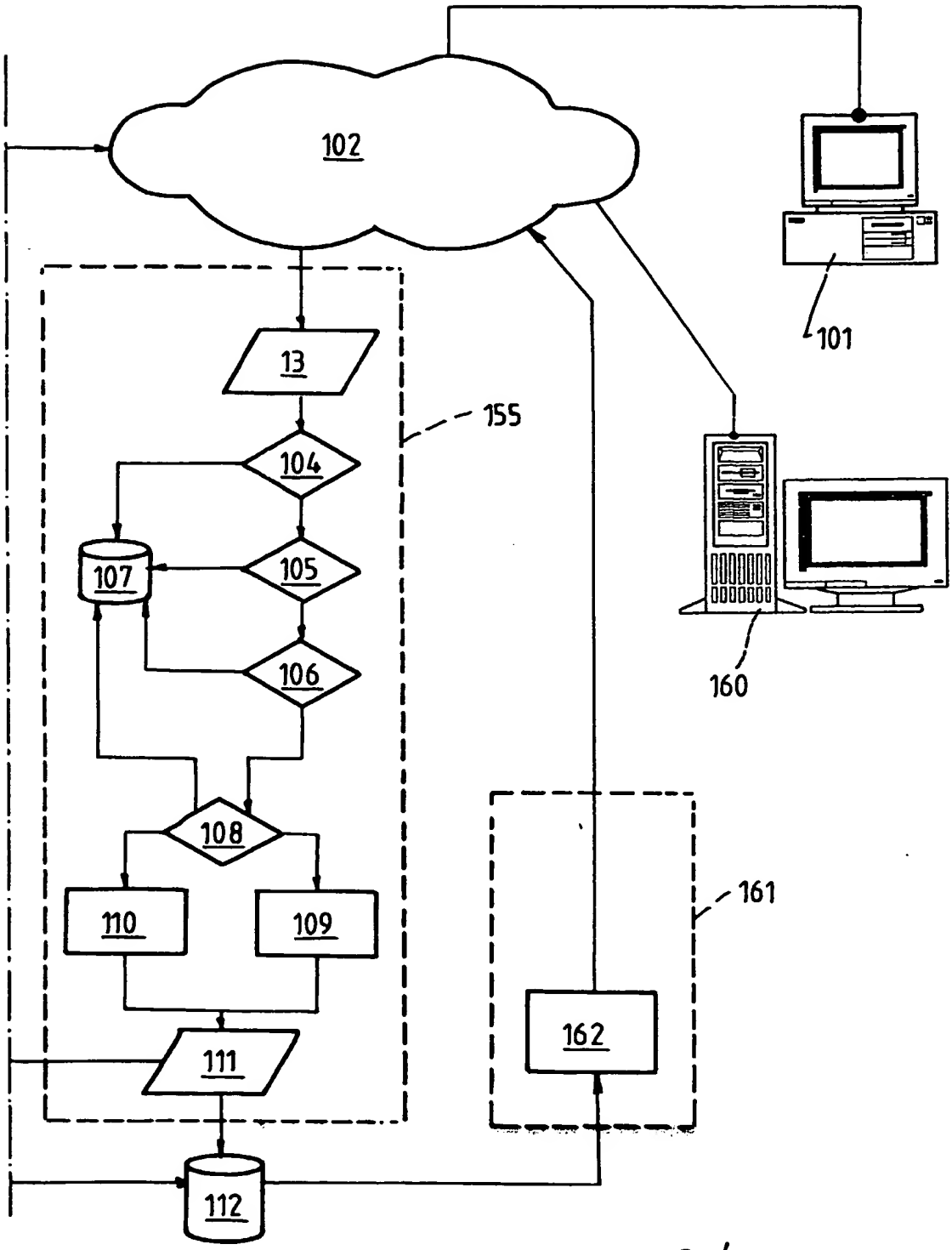


FIG. 28b

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU01/01629

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. [7]: G06F 15/173

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT with keywords

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2000/75827 A (WEBSIDESTORY, INC) 14 December 2000	1 - 37
X	US 6052730 A (FELCIANO et al) 18 April 2000	1 - 37
X	US 5717860 A (GRABER et al) 10 February 1998	1 - 37

[X] Further documents are listed in the continuation of Box C [X] See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 February 2002	22 FEB 2002
Name and mailing address of the ISA/AU	Authorized officer
AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	J.W. THOMSON Telephone No : (02) 6283 2214

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU01/01629

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5712979 A (GRABER et al) 27 January 1998	5 - 23, 34 - 37
X	US 5220655 A (TSUTSUI) 15 June 1993	5 - 23, 34 - 37

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/AU01/01629

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member			
WO	200075827	AU	200043569		
US	6052730	NONE			
US	5717860	AU	72404/96	WO	9711430
US	5712979	NONE			
US	5220655	JP	3260757		

END OF ANNEX

This Page is inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURED OR ILLEGIBLE TEXT OR DRAWING
- ☑ SKEWED/SLANTED IMAGES
- ☐ COLORED OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REPERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning documents *will not* correct images problems checked, please do not report the problems to the IFW Image Problem Mailbox